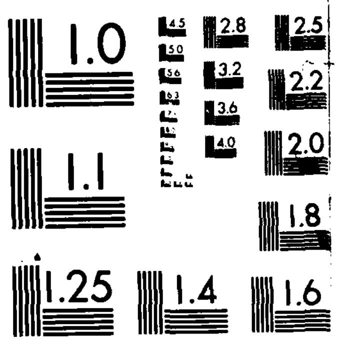
AD-A161 718 MINIMIZATION OF A SIX VARIABLE BOOLEAN FUNCTION(U) 1/2
NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA SYSTEMS
DIRECTORATE L M KOCH MAY 85 NADC-85135-20
UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

REPORT NO. NADC-85135-20

MINIMIZATION OF A SIX VARIABLE BOOLEAN FUNCTION

PREPARED BY:
Lorraine M. Koch
Systems Directorate (Code 2011)
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA 18974

MAY 1985

NOTICES

REPORT NUMBERING SYSTEM — The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteenth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT — The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: T.W. Castaldi DATE: 10/24/85

T.W. CASTALDI
Director, Systems Directorate

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A161718

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS N/A
2a SECURITY CLASSIFICATION AUTHORITY N/A	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NADC-85135-20	5 MONITORING ORGANIZATION REPORT NUMBER(S) N/A
6a NAME OF PERFORMING ORGANIZATION Systems Directorate 6b OFFICE SYMBOL (If applicable) 2011	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) Naval Air Development Center Warminster, PA 18974-5000	7b. ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION N/A 8b OFFICE SYMBOL (If applicable) N/A	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N/A
8c. ADDRESS (City, State, and ZIP Code) N/A	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: N/A; PROJECT NO: N/A; TASK NO: N/A; WORK UNIT ACCESSION NO: N/A

11 TITLE (Include Security Classification)

MINIMIZATION OF A SIX VARIABLE BOOLEAN FUNCTION

12 PERSONAL AUTHOR(S) Lorraine M. Koch

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM N/A TO N/A	1985 May	114

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Digital Design, MSI, Circuits Function Realization, Minimization

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The MINIZE computer program is a minimization model which can be used to aid in the design and optimization of a digital system. An interactive input section allows the user to enter any six variable boolean function. MINIZE will then determine the minimum number of four-input medium-scale integration (MSI) circuits needed to realize that function and output their inputs.

See p 5

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Lorraine M. Koch	22b TELEPHONE (Include Area Code) 215-441-1595 22c OFFICE SYMBOL 2011

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

SECTION I: INTRODUCTION

1.1 BACKGROUND

For many years now digital systems have been designed to perform specific functions. In the earlier years, the primary electronic element was vacuum tubes. Later came the advent of the transistor and recently the integrated circuit. Whether the system be composed of vacuum tubes, transistors, or integrated circuits, one of the major concerns in physically realizing the system is cost. The cost of a particular system equates to the number of logic elements in that system. Therefore, one of the aspects in designing digital systems is the problem of minimization. Given a function f, find the cheapest form f' such that f' denotes f.

Today, the cheapest form equates to the minimum number of integrated circuits, whether they be small-scale integration (SSI) circuits, medium-scale integration (MSI) circuits, large-scale integration (LSI) circuits or even the most recent, very large-scale integration (VLSI) circuits.

The task of realizing a boolean function of six or fewer variables is not unreasonable and can be done by hand using Karnaugh Maps. However, when attempting to minimize the number of integrated circuits required to realize a function with greater than five variables, the task becomes enormous for it involves methods which require a considerable amount of exhaustive searching. Some computer-

assisted tool is necessary to consider all the cases.

1.2 SCOPE OF THE MODEL

A function consisting of four variables can be realized in a single eight-input multiplexer. Three variables are "data select" lines while the fourth variable, or the binary constants 0 and 1, are applied to the eight input lines (I_0 through I_7). This is shown in Figure 1.1. Functions with five or more variables can be realized with multiplexer networks designed in two stages. In the case of a function with six variables, the function can be realized with eight four-input multiplexers (input-stage) feeding into a single eight-input multiplexer (output-stage). Figure 1.2 illustrates this design.

As mentioned previously, Karnaugh Maps can be used to aid in the realization of these functions. However, in the case of the six variable function, it would require evaluating twenty cases. These twenty cases are outlined and explained in Section II of this paper.

MINIZE is a computer program which can be used to aid in the minimization of a six variable combinational boolean function. This program has an interactive input section which allows the user to enter the number of minterms in the function and then the actual minterms (or elements that are equal to one). Output consists of the minimum number of four-input MSI's (input-stage multiplexers) needed to realize this function and their required inputs.

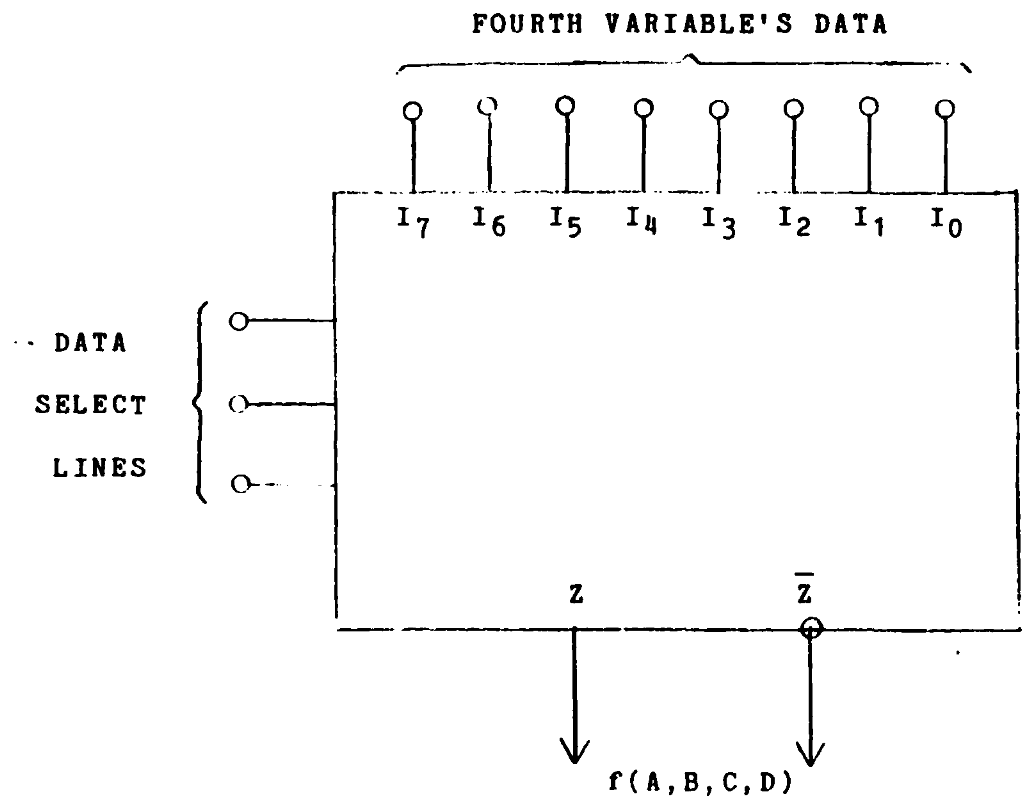


FIGURE 1.1 EIGHT-INPUT MULTIPLEXER USED TO REALIZE A FOUR-VARIABLE FUNCTION.

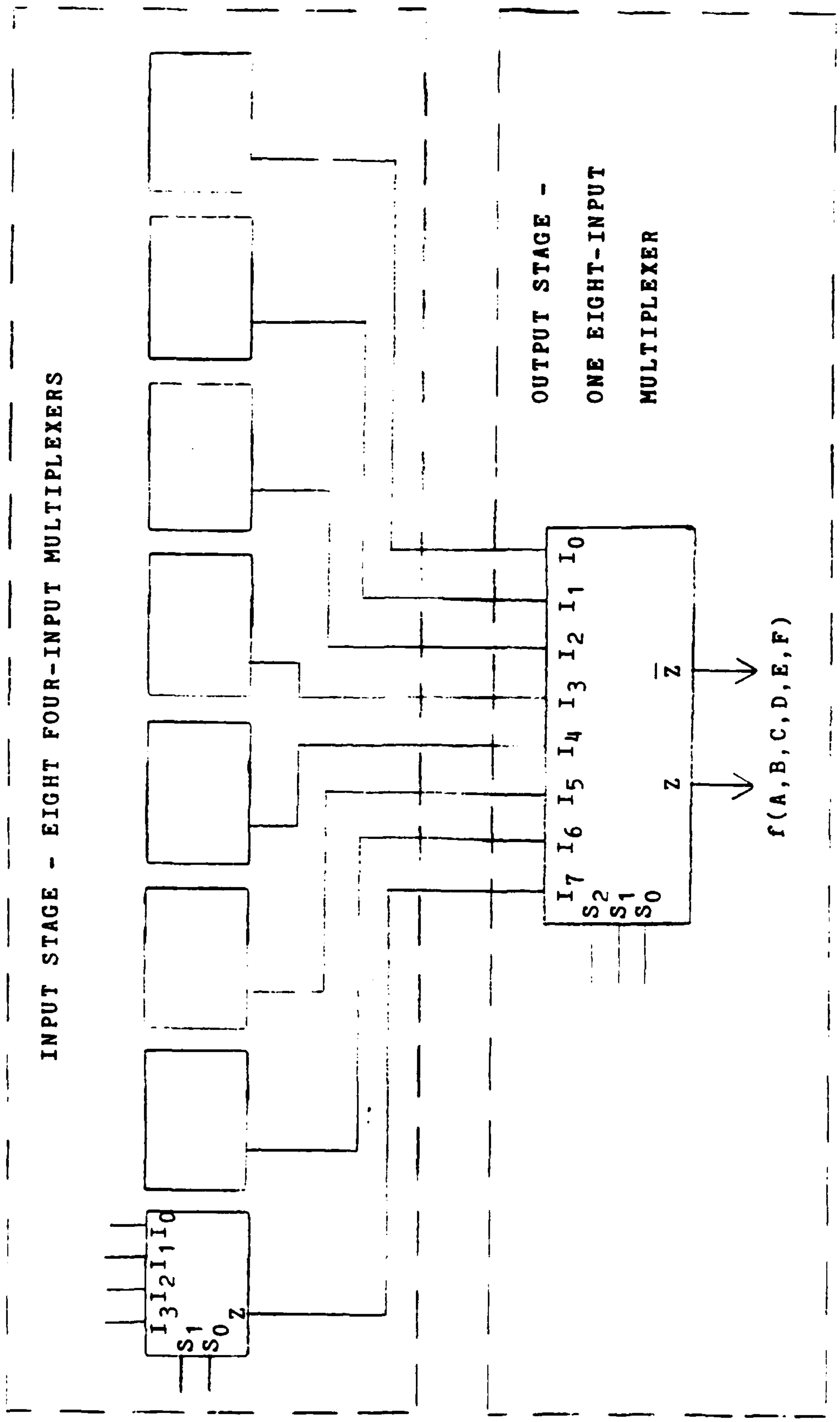


FIGURE 1.2 MULTIPLEXER NETWORK USED TO REALIZE A SIX-VARIABLE FUNCTION.

1.3 PURPOSE OF THE PAPER

The purpose of this paper is to describe the MINIZE program and to provide a user's manual. MINIZE is written in HP Basic for use on a Hewlett-Packard 9845B desktop computer with Advanced Programming capability.

Section ~~II~~ 2 describes the minimization process, along with the techniques used in the model. An example is presented to illustrate the method.

Section ~~III~~ 3 is the user's manual for the MINIZE program. This section outlines the inputs, outputs and operating instructions for the model.

Section ~~IV~~ 4 describes the program and each of its subroutines. Appendix A contains flowcharts of these routines.

A summary of the paper is provided in Section V.

SECTION II: MINIMIZATION METHOD

2.1 OVERVIEW

This section describes the minimization process and the techniques used in the model MINIZE.

2.2 APPROACH

MINIZE uses the following method for reducing a six variable function. The process described here is only a slight modification of one presented in reference (1). This procedure can be described in the following series of steps for the function f(A,B,C,D,E,F):

STEP 1: Divide the function into two parts (g1=g1(A,B,C) and g2=g2(D,E,F)), thus reducing the six-variable function to two three-variable problems.

STEP 2: For each function element which is equal to one (minterm), factor out the variable values A,B,C to obtain the function g2(D,E,F) of three variables.

STEP 3: Collect all terms of g2(D,E,F) corresponding to each unique input combination of A,B,C.

STEP 4: Implement each g2(D,E,F) term. This step is described in more detail in Example 1 presented in Section 2.2.1. This step is the realization of the input-stage.

At this point this function can be realized by eight

four-input input-stage multiplexers feeding into a single eight-input output-stage multiplexer as was shown in Figure 1.2. Now the inputs into the eight input-stage multiplexers must be examined to see if any can be eliminated in an effort to reduce this number of multiplexers.

STEP 5: Check the eight sets of four inputs for the following conditions. For each condition that is met, the required number of multiplexers can be reduced by one.

a) All the inputs in the set equal zero. Then this set can be eliminated.

b) All the inputs in the set equal one. Then this set can be eliminated.

c) Two of the sets have identical inputs. Then one set can be eliminated.

c) Two of the sets have inputs that are complements. Then one set can be eliminated.

STEP 6: Select another permutation of the original function f(A,B,C,D,E,F) and repeat Steps 1-5 until the twenty permutations which provide different solutions of this function have been evaluated. These permutations and the rationale for them are described in Section 2.2.2.

When this process is complete MINIZE will output the minimum number of multiplexers needed and their required inputs in order to realize the given function.

2.2.1 EXAMPLE

The following example illustrates this minimization process.

Example 1: Given the following function and minterms, realize this function using the above minimization process.

f(A,B,C,D,E,F)= m(3,7,12,14,15,19,23,27,28,29, 31,35,39,44,45,46,48,49,50,52, 53,55,56,57,59)

Note: This example will only consider one permutation. The case where g1=g1(A,B,C) and g2=g2(D,E,F).

Table 2.1 is the list of ordered and collected minterms after performing Steps 1 through 3. Note that there are eight values for the function g1=g1(A,B,C) (0-7).

Step 4 is to implement the function g2=g2(D,E,F) for each value of g1(A,B,C). For each group of minterms with the same value of g1(A,B,C) (I value), separate the D and E arguments from F in the associated function g2=g2(D,E,F). Then there are four possible values of the subfunction g2a(D,E) (0-3). An F input must be determined for each of these four values. These F inputs can be found by following the next set of conditions.

A) If two minterms are present for a given g2a=g2a(D,E) (F=0 in one minterm and F=1 in the other), then the F input for this g2a(D,E) value is one (1).

TABLE 2.1 ORDERED AND COLLECTED MINTERMS FOR EXAMPLE 1

FOR PERMUTATION A,B,C, D,E,F

MINTERM	g1(A,b,C)	g2(D,E,F)	I VALUE*
3	0 0 0	0 1 1	0
7	0 0 0	1 1 1	0
12	0 0 1	1 0 0	1
14	0 0 1	1 1 0	1
15	0 0 1	1 1 1	1
19	0 1 0	0 1 1	2
23	0 1 0	1 1 1	2
27	0 1 1	0 1 1	3
28	0 1 1	1 0 0	3
29	0 1 1	1 0 1	3
31	0 1 1	1 1 1	3
35	1 0 0	0 1 1	4
39	1 0 0	1 1 1	4
44	1 0 1	1 0 0	5
45	1 0 1	1 0 1	5
46	1 0 1	1 1 0	5
48	1 1 0	0 0 0	6
49	1 1 0	0 0 1	6
50	1 1 0	0 1 0	6
52	1 1 0	1 0 0	6
53	1 1 0	1 0 1	6
55	1 1 0	1 1 1	6
56	1 1 1	0 0 0	7
57	1 1 1	0 0 1	7
59	1 1 1	0 1 1	7

*value of g1(A,B,C)

B) If no minterms are present for the g2a(D,E), then the F input for the g2a(D,E) value is zero (0).

C) If only one minterm is present for the g2a(D,E) subfunction then:

1) If F=0, then the F input is ~F for this g2a(D,E) value.

2) If F=1, then the F input is F for this g2a(D,E) value.

Table 2.2 presents the F inputs for each g2a(D,E) value under each I value for Example 1.

TABLE 2.2 F INPUTS FOR EXAMPLE 1

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	F
1	0	0	~F	1
2	0	F	0	F
3	0	F	1	F
4	0	F	0	F
5	0	C	1	~F
6	1	~F	1	F
7	1	F	0	0

Step 5 is to reduce this set of inputs. Note that I_0, I_2, and I_4 are all equivalent, so two sets can be eliminated (say I_0 and I_2). Therefore, the minimum number of multiplexers needed to realize this function for this permutation is six (eight minus two). Figure 2.1 graphically illustrates this realization.

2.2.2 PERMUTATIONS PROVIDING UNIQUE SOLUTIONS

This section outlines the twenty permutations that

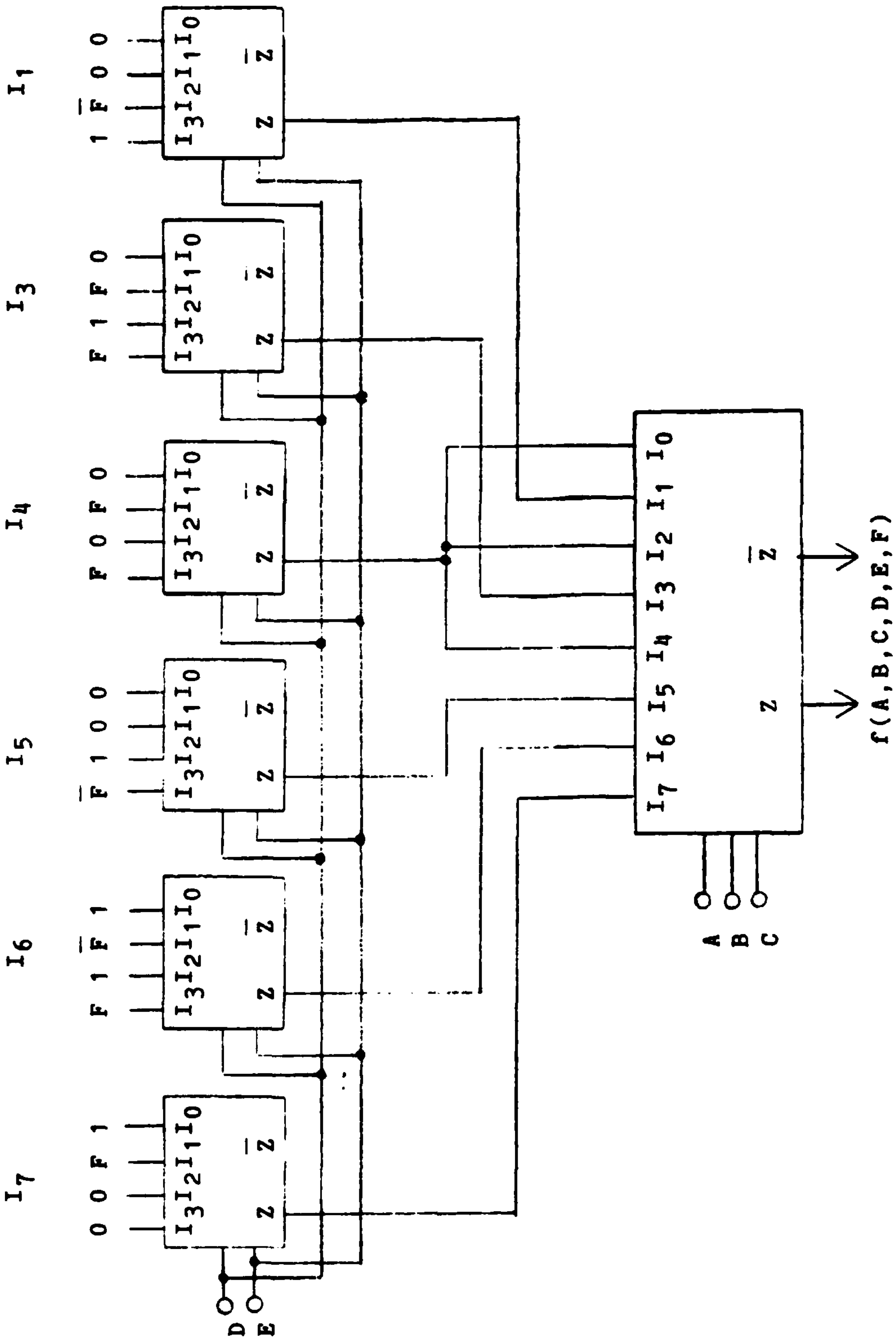


FIGURE 2.1 REALIZATION OF THE FUNCTION IN EXAMPLE 1.

provide unique solutions using this minimization process and presents the rationale behind why these are the only ones that need to be evaluated.

If all permutations of the six variable function were necessary, then there would be 6! or 720 cases to consider. However, consider the following:

1) Say that the last three variables (function g2(D,E,F)) were fixed and were not permuted. Then there would be six permutations of the function f(A,B,C,D,E,F) (the six permutations of the function g1(A,B,C)). When the minterms are grouped and collected by I value (the value of the function g1=g1(A,B,C)) for a particular permutation, then these same minterms would also be grouped together under any of these six permutations. Table 2.3 illustrates this for Example 1 for permutation B,A,C, D,E,F. Since the function g2(D,E,F) was fixed for all these permutations, then the calculated F inputs assigned would still be the same. They would just appear under a different I value. Since this does not affect the reduction process the set of F inputs can be reduced the same for all six permutations and the reduced number of multiplexers required would be the same.

2) Say that the first three elements are fixed (function g1=g1(A,B,C)). There would then be six permutations of the function g2=g2(D,E,F). All

TABLE 2.3 COLLECTED MINTERMS FOR EXAMPLE 1 FOR PERMUTATIONS (A,B,C, D,E,F) AND (B,A,C, D,E,F)

	PERMUTATION (A,B,C, D,E,F)			PERMUTATION (B,A,C, D,E,F)		
MIN-TERMS	g1(A,B,C)	g2(D,E,F)	I VALUE	g1(B,A,C)	g2(D,E,F)	I VALUE
3	0 0 0	0 1 1	0	0 0 0	0 1 1	0
7	0 0 0	1 1 1	0	0 0 0	1 1 1	0
12	0 0 1	1 0 0	1	0 0 1	1 0 0	1
14	0 0 1	1 1 0	1	0 0 1	1 1 0	1
15	0 0 1	1 1 1	1	0 0 1	1 1 1	1
19	0 1 0	0 1 1	2	1 0 0	0 1 1	4
23	0 1 0	1 1 1	2	1 0 0	1 1 1	4
27	0 1 1	0 1 1	3	1 0 1	0 1 1	5
28	0 1 1	1 0 0	3	1 0 1	1 0 0	5
29	0 1 1	1 0 1	3	1 0 1	1 0 1	5
31	0 1 1	1 1 1	3	1 0 1	1 1 1	5
35	1 0 0	0 1 1	4	0 1 0	0 1 1	2
39	1 0 0	1 1 1	4	0 1 0	1 1 1	2
44	1 0 1	1 0 0	5	0 1 1	1 0 0	3
45	1 0 1	1 0 1	5	0 1 1	1 0 1	3
46	1 0 1	1 1 0	5	0 1 1	1 1 0	3
48	1 1 0	0 0 0	6	1 1 0	0 0 0	6
49	1 1 0	0 0 1	6	1 1 0	0 0 1	6
50	1 1 0	0 1 0	6	1 1 0	0 1 0	6
52	1 1 0	1 0 0	6	1 1 0	1 0 0	6
53	1 1 0	1 0 1	6	1 1 0	1 0 1	6
55	1 1 0	1 1 1	6	1 1 0	1 1 1	6
56	1 1 1	0 0 0	7	1 1 1	0 0 0	7
57	1 1 1	0 0 1	7	1 1 1	0 0 1	7
59	1 1 1	0 1 1	7	1 1 1	0 1 1	7

these permutations would also require the same number of multiplexers when reduced. The following four cases explain why this is true.

Case 1: If all eight minterms for a given I value are present, then all F inputs for this I value are one (1). Any permutation of these eight minterms would still produce all eight elements, whose F inputs would also be all ones. In both cases this multiplexer can be eliminated.

Case 2: If no minterms for a given I value are present, then any permutation would still have F inputs all equal to zero. In both cases this multiplexer can be eliminated.

Case 3: If two I values have equivalent F inputs, then any permutation would still result in equivalent inputs. See Example 2a in Table 2.4.

Case 4: If two I values have complementary F inputs, then any permutation would still result in complementary inputs. See Example 2b in Table 2.4.

This means that there are thirty-six cases which result in the same number of required multiplexers (six permutations of g1=g1(A,B,C) X six permutations of g2=g2(D,E,F)). Therefore, the number of permutations of the function f(A,B,C,D,E,F) (divided into two functions

TABLE 2.4 EXAMPLES OF PERMUTATIONS

EXAMPLE 2a: Given the function $f(A,B,C,D,E,F) = \Sigma\, m(18,21,23,42,45,47)$, realize this function for the following permutations: 1) A,B,C, D,E,F and 2) A,B,C, F,D,E.

PERMUTATION 1

g1=g1(A,B,C)	g2=g2(D,E,F)
0 1 0	0 1 0
0 1 0	1 0 1
0 1 0	1 1 1
1 0 1	0 1 0
1 0 1	1 0 1
1 0 1	1 1 1

PERMUTATION 2

g1=g1(A,B,C)	g2=g2(F,D,E)
0 1 0	0 0 1
0 1 0	1 1 0
0 1 0	1 1 1
1 0 1	0 0 1
1 0 1	1 1 0
1 0 1	1 1 1

F INPUTS (Permutation 1)

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	ALL ZEROS			
1	ALL ZEROS			
2	0	$\bar{F}$	F	F
3	ALL ZEROS			
4	ALL ZEROS			
5	0	$\bar{F}$	F	F
6	ALL ZEROS			
7	ALL ZEROS			

F INPUTS (Permutation 2)

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	ALL ZEROS			
1	ALL ZEROS			
2	F	0	0	1
3	ALL ZEROS			
4	ALL ZEROS			
5	F	0	0	1
6	ALL ZEROS			
7	ALL ZEROS			

In both cases I_2 and I_5 are equivalent and only one multiplexer is needed to realize this function.

TABLE 2.4 EXAMPLES OF PERMUTATIONS (CONTINUED)

EXAMPLE 2b: Given the function $f(A,B,C,D,E,F) = \Sigma m(17,20,22, 23,40,42,43,45)$, realize this function for the following permutations: 1) A,B,C, D,E,F and 2) A,B,C, F,D,E.

PERMUTATION 1		PERMUTATION 2	
g1=g1(A,B,C)	g2=g2(D,E,F)	g1=g1(A,B,C)	g2=g2(F,D,E)
0 1 0	0 0 1	0 1 0	1 0 0
0 1 0	1 0 0	0 1 0	0 1 0
0 1 0	1 1 0	0 1 0	0 1 1
0 1 0	1 1 1	0 1 0	1 1 1
1 0 1	0 0 0	1 0 1	0 0 0
1 0 1	0 1 0	1 0 1	0 0 1
1 0 1	0 1 1	1 0 1	1 0 1
1 0 1	1 0 1	1 0 1	1 1 0

F INPUTS (Permutation 1)

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL ZEROS			
1	ALL ZEROS			
2	F	0	$\bar{F}$	1
3	ALL ZEROS			
4	ALL ZEROS			
5	$\bar{F}$	1	F	0
6	ALL ZEROS			
7	ALL ZEROS			

F INPUTS (Permutation 2)

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL ZEROS			
1	ALL ZEROS			
2	0	1	$\bar{F}$	F
3	ALL ZEROS			
4	ALL ZEROS			
5	1	0	F	$\bar{F}$
6	ALL ZEROS			
7	ALL ZEROS			

In both cases I_2 and I_5 are complements and only one multiplexer is needed to realize this function.

g1=g1(A,B,C) and g2=g2(D,E,F)) which result in different solutions to this minimization process is twenty (720/36). Table 2.5 lists these permutations.

Note that combination groups such as (A,B,C, D,E,F) and (D,E,F, A,B,C) do not result in the same number of multiplexers when they are reduced. Tables 2.6 and 2.7 perform the minimization process on Example 1 presented in Section 2.2.1 for the permutation where g1=g1(D,E,F) and g2=g2(A,B,C). In this case the set of F inputs for I_0 and I_4 are equivalent. Therefore, this particular permutation can be reduced by one multiplexer, whereas the first permutation considered in Table 2.1 was reduced by two multiplexers.

Appendix B contains a complete output for Example 1.

2.3 SUMMARY

This section describes the minimization method implemented in the MINIZE computer program. This process realizes a six variable boolean function in a minimum number of multiplexers. An example function is examined to aid the reader in understanding the process.

Also presented is a description of and rationale for the twenty permutations or cases that the program must consider in order to determine the minimum.

TABLE 2.5 PERMUTATIONS PROVIDING UNIQUE SOLUTIONS

NUMBER	PERMUTATION	NUMBER	PERMUTATION
1	ABC DEF	11	DEF ABC
2	ABD CEF	12	CEF ABD
3	ABE CDF	13	CDF ABE
4	ABF CDE	14	CDE ABF
5	ACD BEF	15	BEF ACD
6	ACE BDF	16	BDF ACE
7	ACF BDE	17	BDE ACF
8	ADE BCF	18	BCF ADE
9	ADF BCE	19	BCE ADF
10	AEF BCD	20	BCD AEF

TABLE 2.6 ORDERED MINTERMS OF EXAMPLE 1 FOR

PERMUTATION (D,E,F, A,B,C)

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
DEFABC				
	000011	011000	000110	0
	000111	111000	000111	0
	001100	100001	001110	1
	001110	110001	001111	1
	001111	111001	010110	2
	010011	011010	011000	3
	010111	111010	011010	3
	011011	011011	011011	3
	011100	100011	011100	3
	011101	101011	011111	3
	011111	111011	100001	4
	100011	011100	100011	4
	100111	111100	100101	4
	101100	100101	100110	4
	101101	101101	101011	5
	101110	110101	101101	5
	110000	000110	101110	5
	110001	001110	110001	6
	110010	010110	110101	6
	110100	100110	111000	7
	110101	101110	111001	7
	110111	111110	111010	7
	111000	000111	111011	7
	111001	001111	111100	7
	111011	011111	111110	7

TABLE 2.7 F INPUTS FOR EXAMPLE 1 FOR PERMUTATION (D,E,F, A,B,C)

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	0	0	1
1	0	0	0	1
2	0	0	0	~F
3	~F	1	~F	F
4	F	F	F	~F
5	0	F	F	~F
6	F	0	F	0
7	1	1	~F	~F

SECTION III: USER'S MANUAL

3.1 OVERVIEW

This section describes the inputs and outputs of the MINIZE computer program. Instructions for operating the model are also presented.

3.2 INPUTS

The MINIZE program is an interactive program which prompts the user with questions on each of the necessary inputs or options available. Table 3.1 lists all these questions. A comment section in this table is provided to describe the options available to the user and any limitations on the responses.

3.3 OUTPUTS

The program's output begins with a printed hard copy of the entered minterms and their associated binary equivalent. Table 3.2 is an example of this printout for the function in Example 1.

At the end of the input section, the program will ask if the user wishs a list of the twenty permutations. If the user answers yes (Y), the program will print these permutations either on the CRT or the thermal printer, depending on the user-selected printout option. Table 2.5 is an example of this printout.

If the slow detailed version has been selected the program will print three tables for each of the twenty

TABLE 3.1 LIST OF INPUT QUESTIONS

QUESTION NUMBER	QUESTION	COMMENT
Q-1	Enter the number of minterms in this six variable function:	a) Integer b) $1 \leq$ Number ≤ 64
Q-2	Enter the minterms:	a) Integers b) $0 \leq$ Minterm ≤ 63 c) Each minterm must be unique Note: Minterms are then printed on the CRT
Q-3	Are these correct (Y or N)?	a) Y-if minterms correct b) N-if not correct (will return user to Q-1)
Q-4	Do you want to run the fast version or the slow detailed printout version (F=Fast or S=Slow)?	a) F-prints only current permutation number & minimum number of multi-plexers required to date on the CRT (questions continue with Q-5)
	If "S" is selected for Q-4, then: Q-4a Do you want the detailed printout to appear on the CRT or do you want a hard copy (16=CRT 0=hard copy)?	b) S-prints inter-mediate results for each permutation and notes minimum to date on selected printout option: 16=CRT 0=Thermal printer
Q-5	Do you want to list the permutations (Y or N)?	a) Y-lists the 20 permutations b) N-does not
	At the end of the program, after the current function has be evaluated:	
Q-6	Do you wish to evaluate another function (Y or N)?	a) Y-returns user to Q-1 b) N-ends program

TABLE 3.2 SAMPLE PRINTOUT OF THE ORIGINAL MINTERMS WITH THEIR BINARY EQUIVALENT FOR EXAMPLE 1

LIST OF MINTERMS

ORIGINAL MINTERM	BINARY EQUIVALENT
3	000011
7	000111
12	001100
14	001110
15	001111
19	010011
23	010111
27	011011
28	011100
29	011101
31	011111
35	100011
39	100111
44	101100
45	101101
46	101110
48	110000
49	110001
50	110010
52	110100
53	110101
55	110111
56	111000
57	111001
59	111011

permutations. These tables will be printed on the user-selected printout option. Tables 3.3 through 3.5 contain sample printouts of each of these tables for the permutation A,B,C, D,E,F in Example 1.

The first table (Table 3.3) lists the following information:

1) the current permutation
2) the original minterms
3) the minterms after this permutation
4) the sorted minterms after this permutation (in ascending order), and
5) the I value for the rearranged minterm.

This table is then followed by the unreduced F inputs table for this permutation (Table 3.4). The third table for the permutation is then the reduced F inputs (Table 3.5).

If the fast output version is selected, the program displays on the CRT the permutation number that the program is currently evaluating, and the permutation number that so far can be realized in the minimum number of multiplexers. This minimum number of multiplexers required to date is also displayed. This display also appears on the CRT if the slow version is selected.

After all twenty permutations have been evaluated, MINIZE will print on the thermal printer the permutation that reduces to the minimum number of multiplexers necessary in order to realize the given function. The required inputs for this case are also printed. Table 3.6 is an example of

TABLE 3.3 SAMPLE PRINTOUT OF MINTERMS FOR PERMUTATION A,B,C, D,E,F FOR EXAMPLE 1

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ABCDEF				
	000011	000011	000011	0
	000111	000111	000111	0
	001100	001100	001100	1
	001110	001110	001110	1
	001111	001111	001111	1
	010011	010011	010011	2
	010111	010111	010111	2
	011011	011011	011011	3
	011100	011100	011100	3
	011101	011101	011101	3
	011111	011111	011111	3
	100011	100011	100011	4
	100111	100111	100111	4
	101100	101100	101100	5
	101101	101101	101101	5
	101110	101110	101110	5
	110000	110000	110000	6
	110001	110001	110001	6
	110010	110010	110010	6
	110100	110100	110100	6
	110101	110101	110101	6
	110111	110111	110111	6
	111000	111000	111000	7
	111001	111001	111001	7
	111011	111011	111011	7

Note that in this case the minterms after this permutation are equal to the original minterms. Table 2.6 is another example of this type printout for permutation D,E,F, A,B,C.

TABLE 3.4 SAMPLE PRINTOUT OF THE UNREDUCED F INPUTS

TABLE FOR PERMUTATION (A,B,C, D,E,F) FOR EXAMPLE 1

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	F	0	F
1	0	0	¯F	1
2	0	F	0	F
3	0	F	1	F
4	0	F	0	F
5	0	0	1	¯F
6	1	¯F	1	F
7	1	F	0	0

TABLE 3.5 SAMPLE PRINTOUT OF THE REDUCED F INPUTS

TABLE FOR PERMUTATION (A,B,C, D,E,F) FOR EXAMPLE 1

REDUCED F MATRIX MULTIPLEXERS= 6

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT		TO	I4
1	0	0	¯F	1	
2	THIS	EQUIVALENT		TO	I4
3	0	F	1	F	
4	0	F	0	F	
5	0	0	1	¯F	
6	1	¯F	1	F	
7	1	F	0	0	

TABLE 3.6 SAMPLE PRINTOUT OF THE MINIMUM FOR THE FUNCTION IN EXAMPLE 1

MINIMUM MULTIPLEXERS NEEDED IS 5

IT IS PERMUTATION NUMBER 20 WHICH IS THE FOLLOWING:

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BCDAEF				
	000011	000011	000011	0
	000111	001011	000111	0
	001100	011000	001011	1
	001110	011010	001111	1
	001111	011011	011000	3
	010011	100011	011010	3
	010111	101011	011011	3
	011011	110011	011100	3
	011100	111000	011101	3
	011101	111001	011110	3
	011111	111011	100011	4
	100011	000111	100100	4
	100111	001111	100101	4
	101100	011100	100110	4
	101101	011101	101011	5
	101110	011110	101100	5
	110000	100100	101101	5
	110001	100101	101111	5
	110010	100110	110011	6
	110100	101100	110100	6
	110101	101101	110101	6
	110111	101111	110111	6
	111000	110100	111000	7
	111001	110101	111001	7
	111011	110111	111011	7

THE INPUTS INTO THESE MULTIPLEXERS ARE AS FOLLOWS:

REDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	THIS EQUIVALENT TO I1			
1	0	F	0	F
2	ALL VALUES = ZERO			
3	¯F	1	1	¯F
4	0	F	1	¯F
5	THIS EQUIVALENT TO I6			
6	0	F	1	F
7	1	F	0	0

this printout for Example 1.

Appendix B contains a full detailed printout of all permutations for Example 1.

3.4 OPERATING INSTRUCTIONS

In order to run the MINIZE program, the user must perform the following steps on the HP9845B desktop computer:

1) Place the program tape in the right-hand tape reader.
2) Type LOAD "MINIZE" and press the EXECUTE key.
3) When the tape stops, press the RUN key.
4) Answer the prompted input questions.

The program can be stopped during execution by pressing the STOP key. To begin completely over again, press the STOP key twice and then the RUN key.

SECTION IV: PROGRAM DESCRIPTION

4.1 OVERVIEW

This section describes the program MINIZE by providing an explanation of each of the subroutines.

Flowcharts of most of the routines can be found in Appendix A. Some routines are very short, so only an explanation of their purpose is given in this section. A complete listing of the program is provided in Appendix C.

4.2 PROGRAM MINIZE

This is the executive routine. It controls the calling order of all the subroutines and, at the end, prints the permutation which can be realized in a minimum number of 4-input multiplexers. It also prints the required inputs into these multiplexers.

4.3 SUBROUTINE INTRO

This routine introduces the user to the MINIZE program by printing the following on the CRT:

```
                         M I N I Z E

     THIS PROGRAM WILL REALIZE A SIX VARIABLE COMBINATIONAL
BOOLEAN FUNCTION F(A,B,C,D,E,F) IN A MINIMUM NUMBER OF
4-INPUT MULTIPLEXERS.  YOU WILL BE ASKED FOR THE NUMBER OF
MINTERMS (OR ELEMENTS) AND THEN WHAT THESE MINTERMS ARE.


     MINIZE WILL OUTPUT TO YOU THE MINIMUM NUMBER OF
4-INPUT MULTIPLEXERS YOU WILL NEED AND WHAT EACH OF THE
INPUTS ARE INTO EACH MULTIPLEXER.


PRESS   CONT   WHEN YOU ARE READY.
```

4.4 SUBROUTINE READ-DATA

This routine initializes various arrays, variables and flags used throughout MINIZE. Table 4.1 lists some of these variables and arrays and their purpose.

4.5 SUBROUTINE GET-INPUTS

Subroutine Get-inputs prompts the user with questions on each of the required inputs and options available. Table 3.1 listed all of these questions.

Certain checks are made on some of the inputs, and as a result an error message may appear on the CRT. The flowchart of this subroutine in Appendix A has labeled the three different error messages by number. They are as follows:

Error Message #1: "NUMBER OF MINTERMS MUST BE BETWEEN 1 AND 64 INCLUSIVE."

Error Message #2: "MINTERMS MUST BE BETWEEN 0 AND 63 INCLUSIVE."

Error Message #3: "EACH MINTERM MUST BE UNIQUE."

4.6 SUBROUTINE ASSIGN-F-VALUES

This subroutine creates the unreduced F table by assigning all the F inputs (one for each g2a(D,E) index under each I value). This is Step 4 in the minimization process, which is outlined in Section 2.2 and is described in more detail in Section 2.2.1.

TABLE 4.1 PARTIAL LIST OF DATA ARRAYS AND VARIABLES

NAME	ARRAY/VARIABLE	TYPE	PURPOSE
Perm	array	Integer	The 20 permutations to be evaluated (numeric values)
Alpha-perm$	array	Alpha-numeric	The alpha-numeric values of the 20 permutations
Dividend	array	Integer	The powers of 2 necessary to calculate the binary equivalents
All1$	variable	Alpha-numeric	Prints "ALL VALUES = ONE" message
All0$	variable	Alpha-numeric	Prints "ALL VALUES = ZERO" message
Equiv$	variable	Alpha-numeric	Prints "THIS EQUIVALENT TO" message
Compl$	variable	Alpha-numeric	Prints "THIS COMPLEMENT OF" message
Numb$	array	Alpha-numeric	Prints "I0" through "I7" for equiv/comple messages
Zero$	variable	Alpha-numeric	Prints " 0 "
One$	variable	Alpha-numeric	Prints " 1 "
Defalt$	variable	Alpha-numeric	Prints " " (blanks)
Inp-f$	variable	Alpha-numeric	Prints " F "
Inp-nf$	variable	Alpha-numeric	Prints " ¯F "
Min-mult	variable	Integer	Stores minimum number of multiplexers to date. Initialized to 9.
Min-where	variable	Integer	Stores minimum permutation number to date. Initialized to 0.

4.7 SUBROUTINE REDUCE

The unreduced F table is examined in this subroutine to determine if any of the multiplexers can be eliminated. This is Step 5 in the minimization process and is explained in Section 2.2. After examining the unreduced F table and creating the reduced F table, this subroutine then checks to see if this permutation requires fewer multiplexers then the minimum to date. If it does, this permutation and its reduced F inputs are then stored as the new minimum to date.

SECTION V: SUMMARY

This paper describes the MINIZE computer program, which is a minimization model that can be used to aid in the design and optimization of a digital system. The model is written in HP Basic for use on a Hewlett-Packard 9845B desktop computer with Advanced Programming capability.

An interactive input section allows the user to enter any six variable boolean function. MINIZE then evaluates this function according to the methods and procedures of a minimization process which realizes this function in a multiplexer network designed in two stages. The input-stage normally requires eight four-input multiplexers which then feed into an output-stage consisting of a single eight-input multiplexer. This computer model will attempt to reduce the number of input-stage multiplexers required.

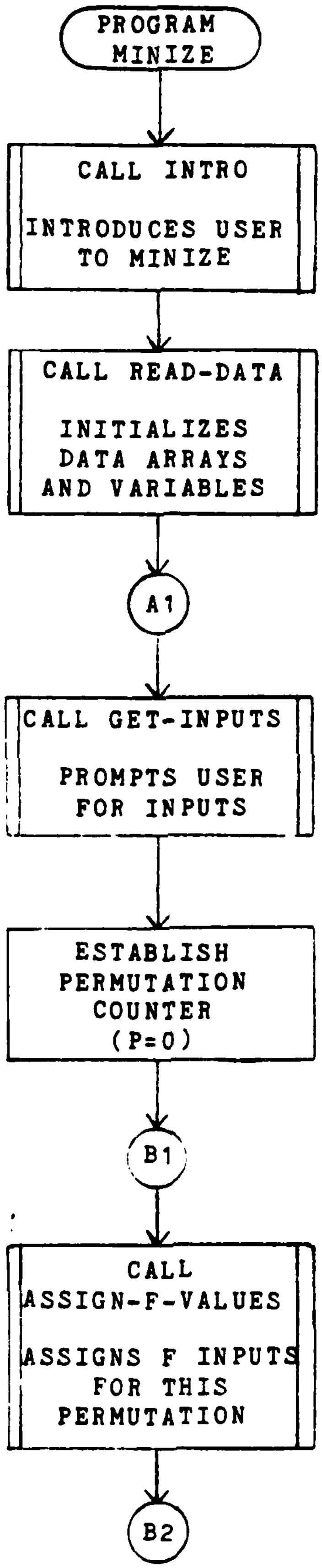
A description of the minimization process, along with a program description, user's guide, flowchart, listing and sample printout are all provided to aid both the reader and user.

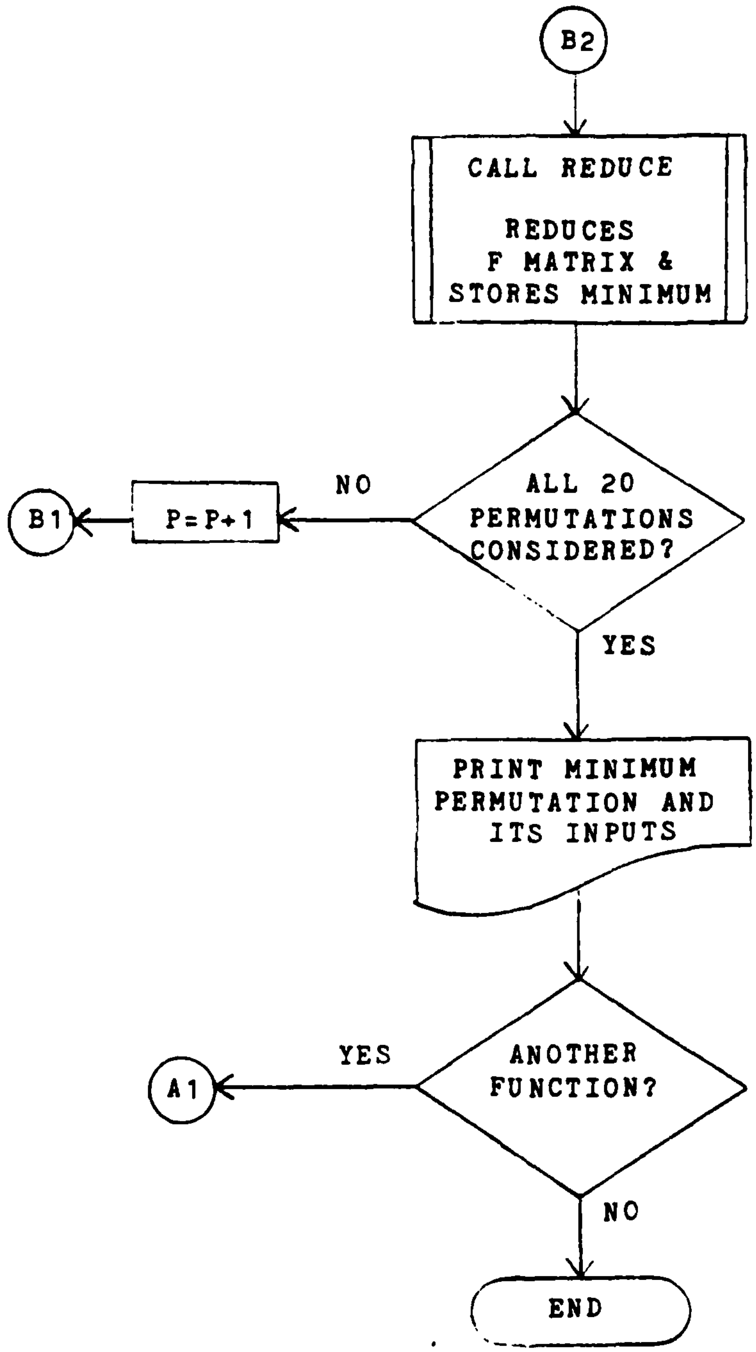
REFERENCES

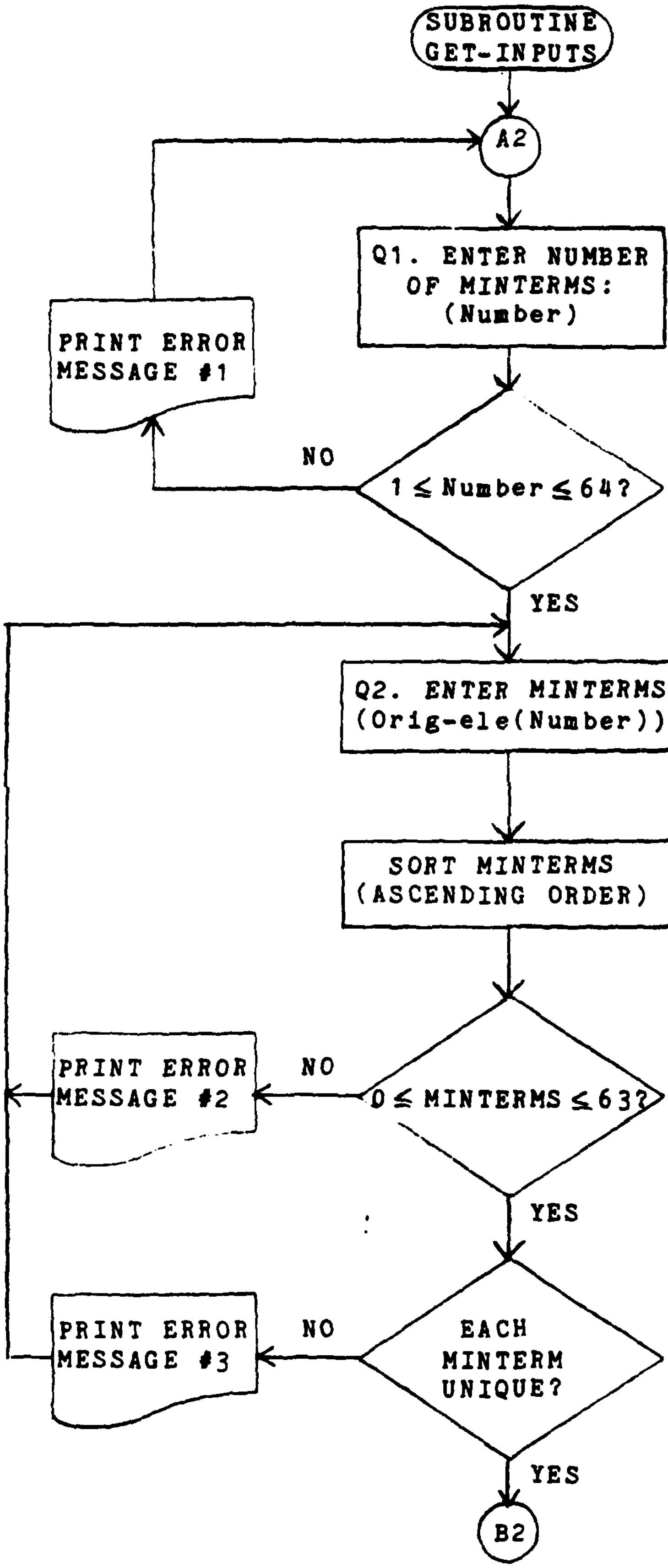
(1) Barna, Arpad and Porat, Dan I., Integrated Circuits in Digital Electronics, John Wiley & Sons, New York, 1973.

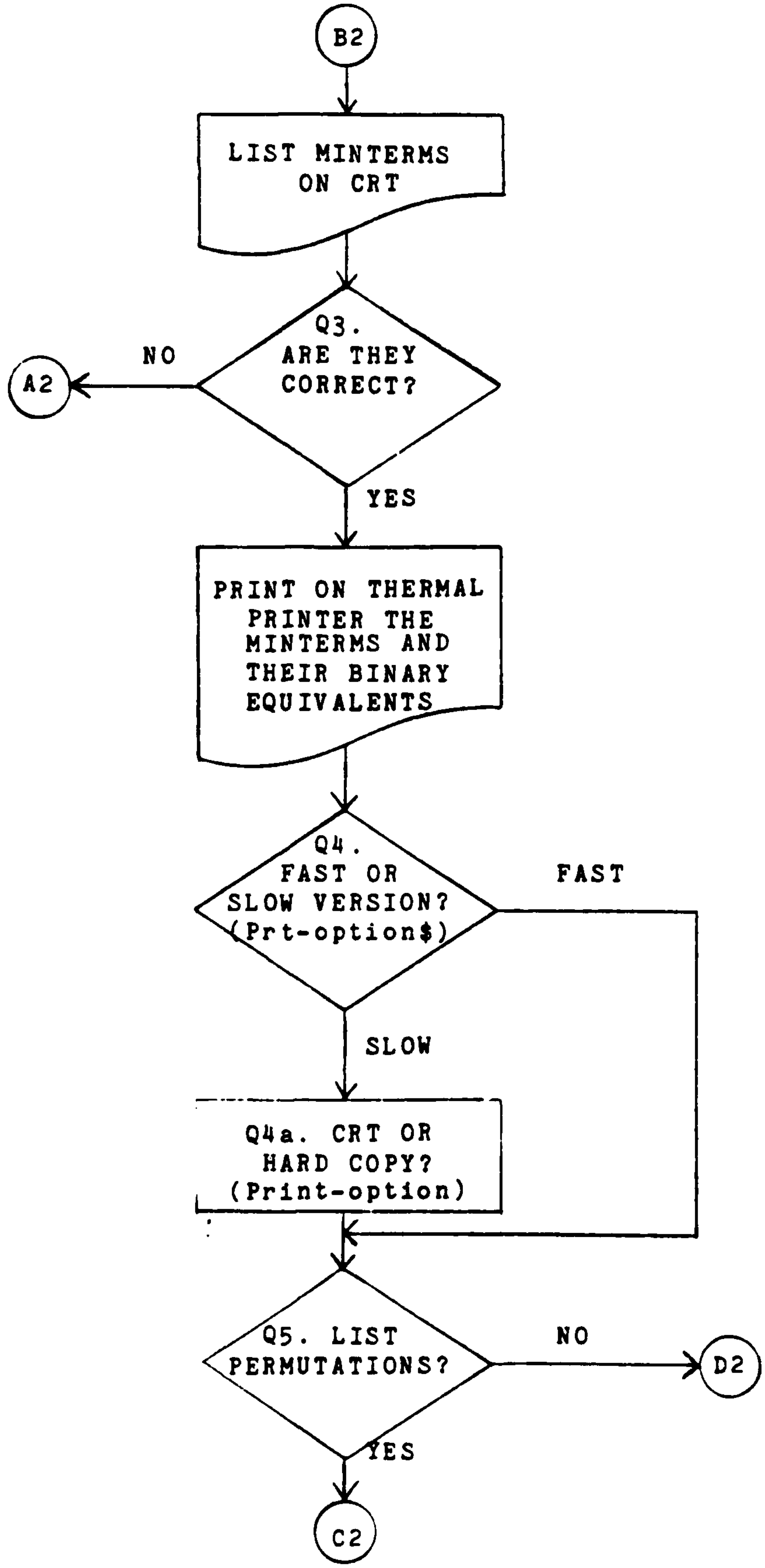
APPENDIX A:

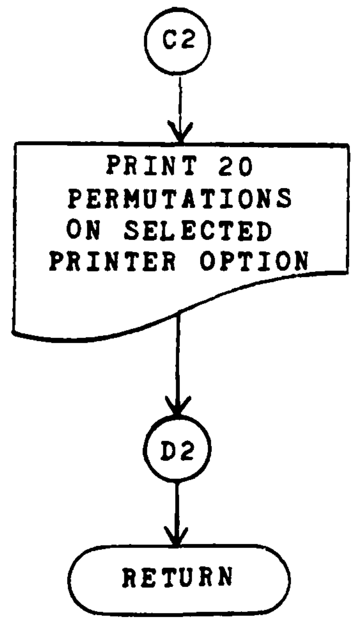
FLOWCHARTS FOR MINIZE

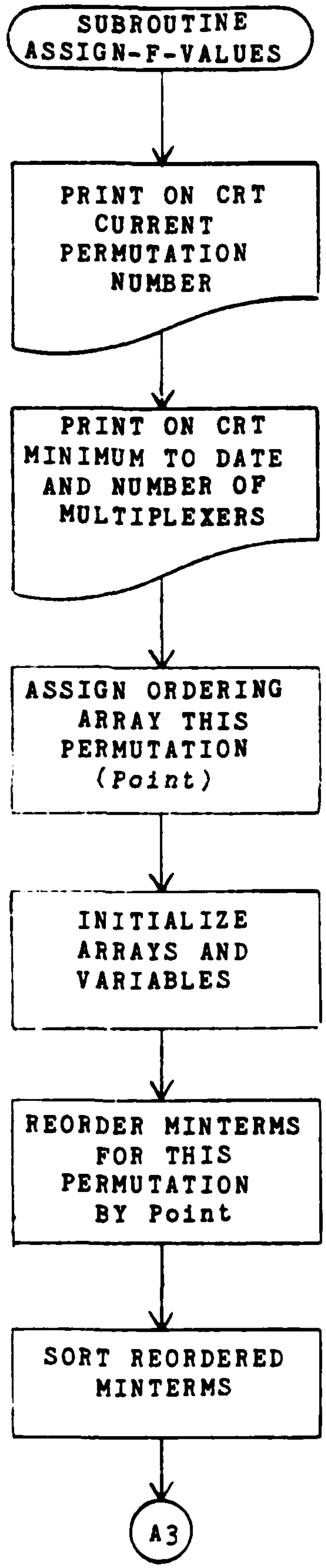

PROGRAM MINIZE
CALL INTRO
INTRODUCES USER TO MINIZE
CALL READ-DATA
INITIALIZES DATA ARRAYS AND VARIABLES
A1
CALL GET-INPUTS
PROMPTS USER FOR INPUTS
ESTABLISH PERMUTATION COUNTER (P=0)
B1
CALL ASSIGN-F-VALUES
ASSIGNS F INPUTS FOR THIS PERMUTATION
B2

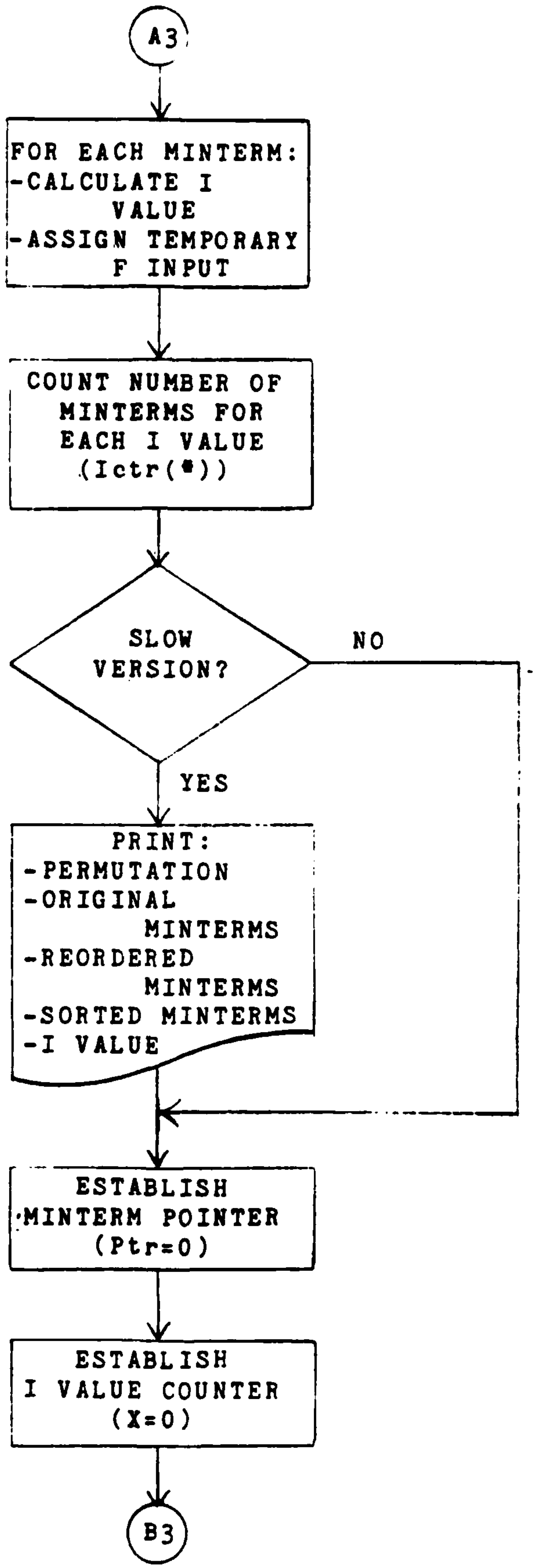

B2
CALL REDUCE
REDUCES
F MATRIX &
STORES MINIMUM
ALL 20
PERMUTATIONS
CONSIDERED?
NO
P=P+1
B1
YES
PRINT MINIMUM
PERMUTATION AND
ITS INPUTS
ANOTHER
FUNCTION?
YES
A1
NO
END

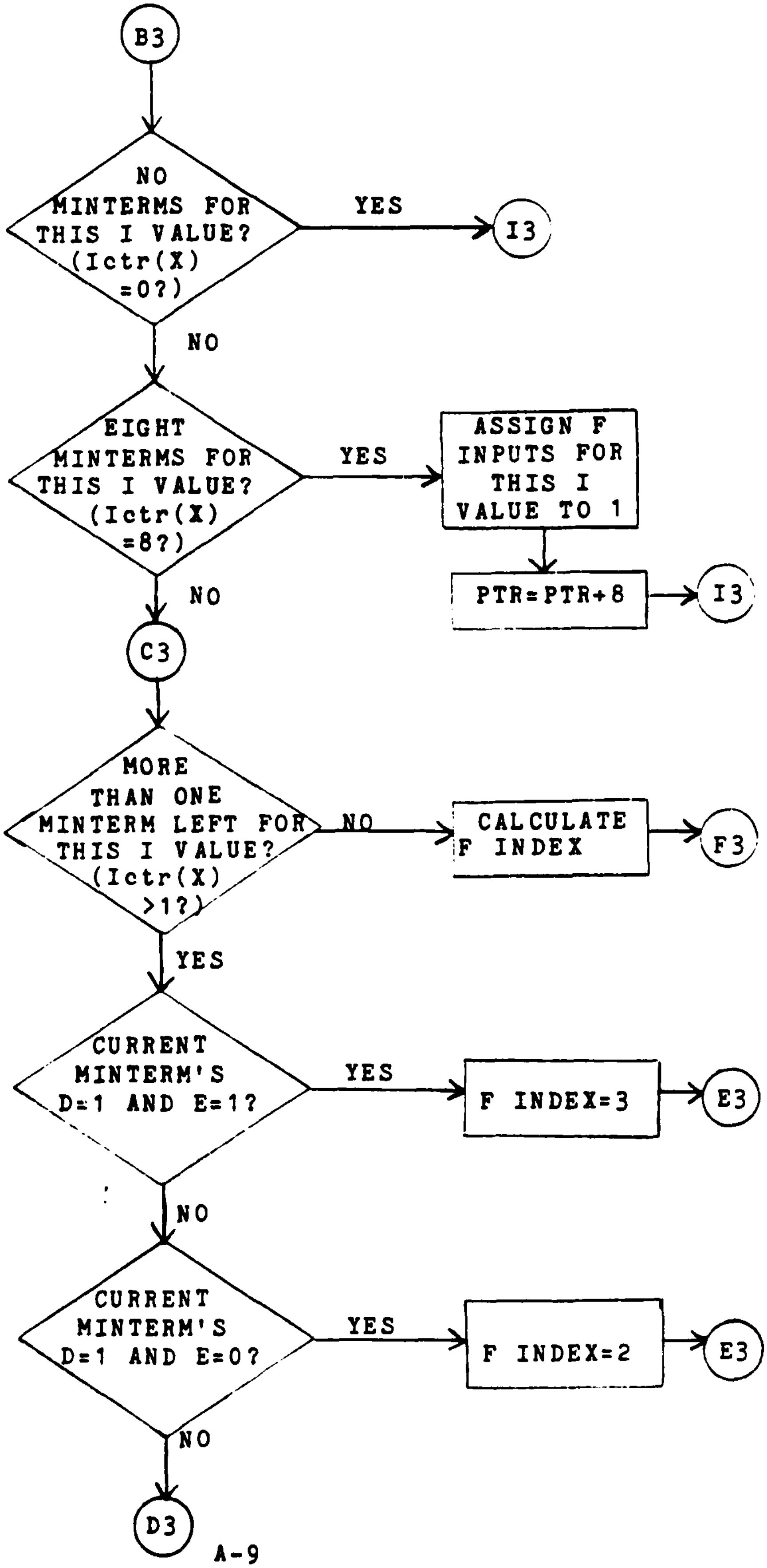

SUBROUTINE GET-INPUTS
A2
Q1. ENTER NUMBER OF MINTERMS: (Number)
PRINT ERROR MESSAGE #1
NO
1 ≤ Number ≤ 64?
YES
Q2. ENTER MINTERMS (Orig-ele(Number))
SORT MINTERMS (ASCENDING ORDER)
PRINT ERROR MESSAGE #2
NO
0 ≤ MINTERMS ≤ 63?
YES
PRINT ERROR MESSAGE #3
NO
EACH MINTERM UNIQUE?
YES
B2

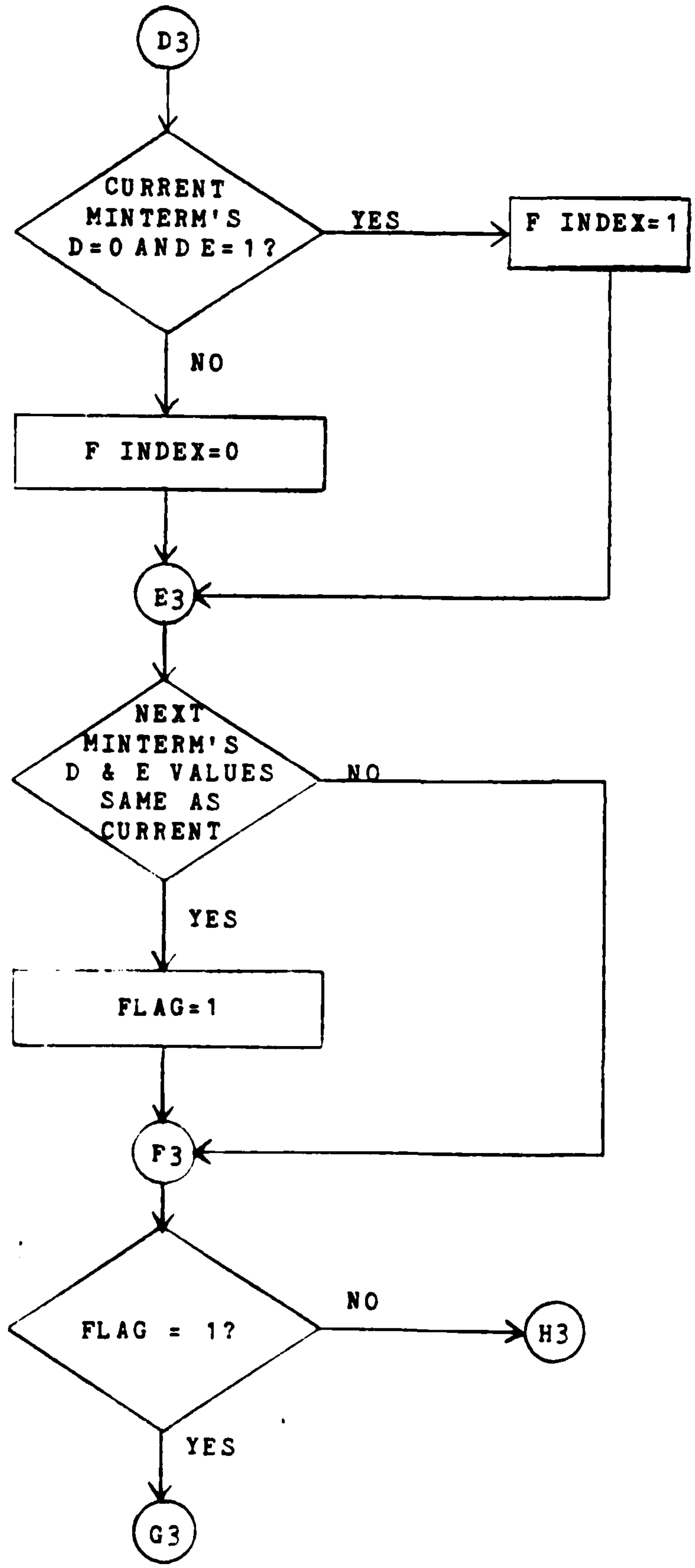

B2
LIST MINTERMS ON CRT
Q3. ARE THEY CORRECT?
NO
A2
YES
PRINT ON THERMAL PRINTER THE MINTERMS AND THEIR BINARY EQUIVALENTS
Q4. FAST OR SLOW VERSION? (Prt-option$)
FAST
SLOW
Q4a. CRT OR HARD COPY? (Print-option)
Q5. LIST PERMUTATIONS?
NO
D2
YES
C2

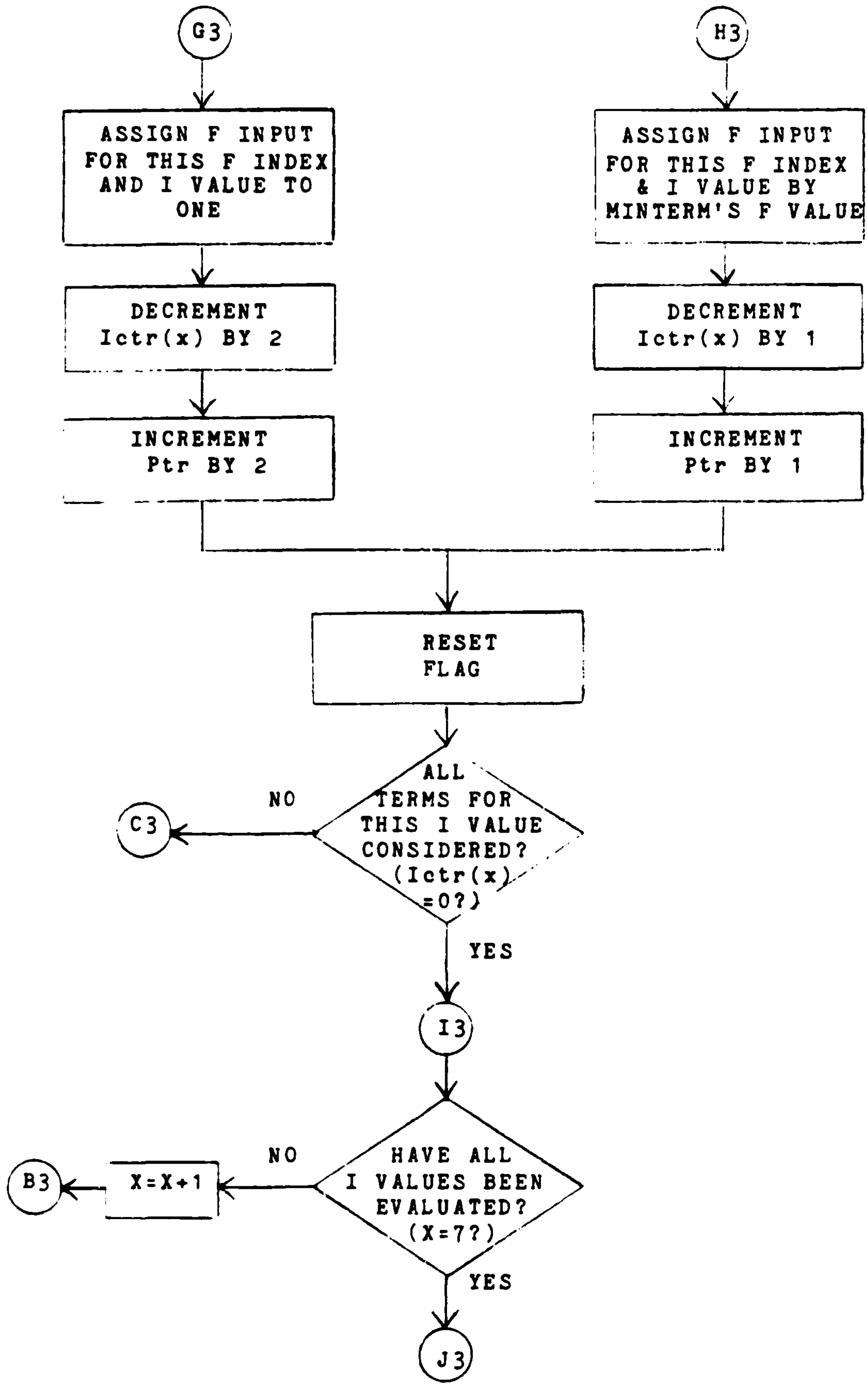

C2
PRINT 20
PERMUTATIONS
ON SELECTED
PRINTER OPTION
D2
RETURN

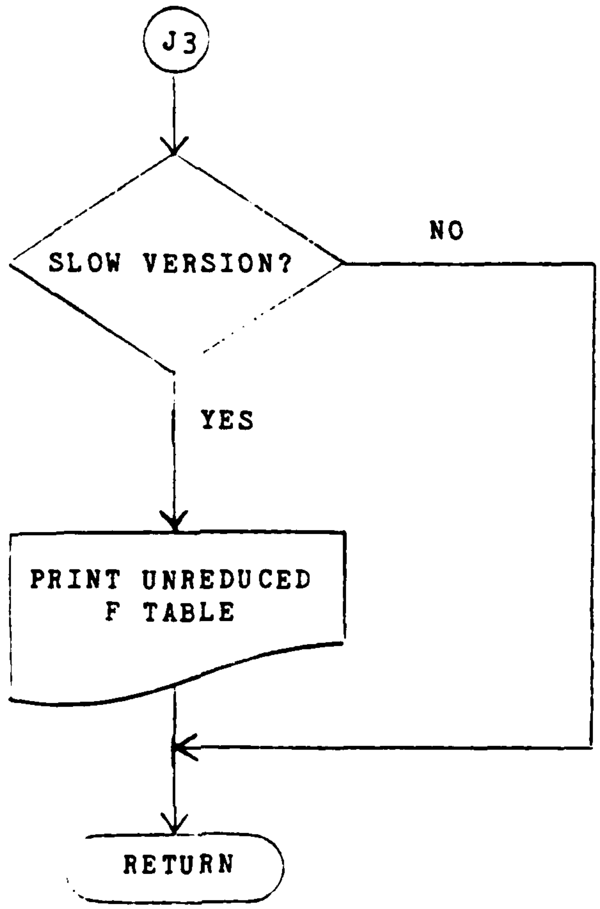

SUBROUTINE ASSIGN-F-VALUES
PRINT ON CRT CURRENT PERMUTATION NUMBER
PRINT ON CRT MINIMUM TO DATE AND NUMBER OF MULTIPLEXERS
ASSIGN ORDERING ARRAY THIS PERMUTATION (Point)
INITIALIZE ARRAYS AND VARIABLES
REORDER MINTERMS FOR THIS PERMUTATION BY Point
SORT REORDERED MINTERMS
A3

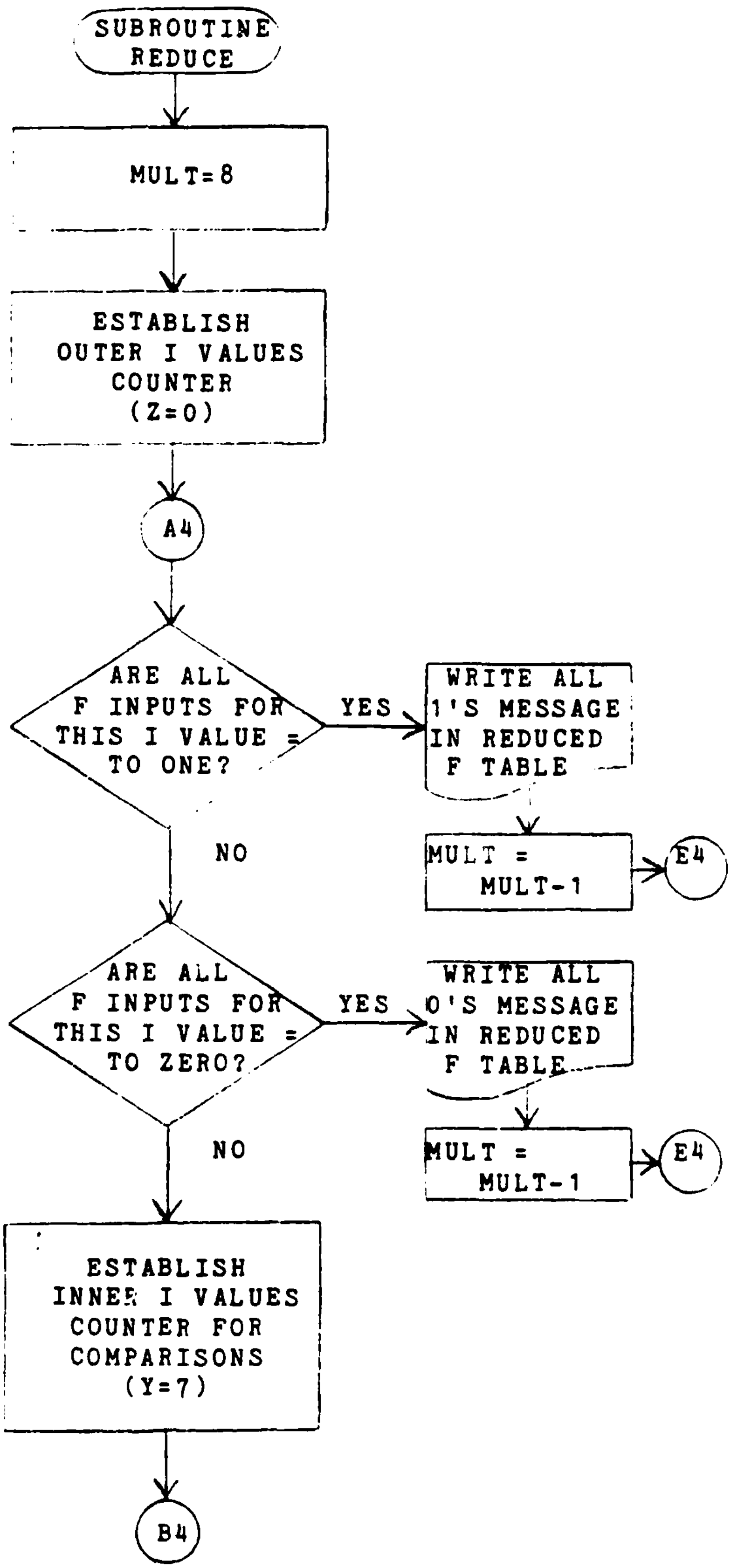

A3
FOR EACH MINTERM:
-CALCULATE I VALUE
-ASSIGN TEMPORARY F INPUT
COUNT NUMBER OF MINTERMS FOR EACH I VALUE (Ictr(*))
SLOW VERSION?
NO
YES
PRINT:
-PERMUTATION
-ORIGINAL MINTERMS
-REORDERED MINTERMS
-SORTED MINTERMS
-I VALUE
ESTABLISH MINTERM POINTER (Ptr=0)
ESTABLISH I VALUE COUNTER (X=0)
B3

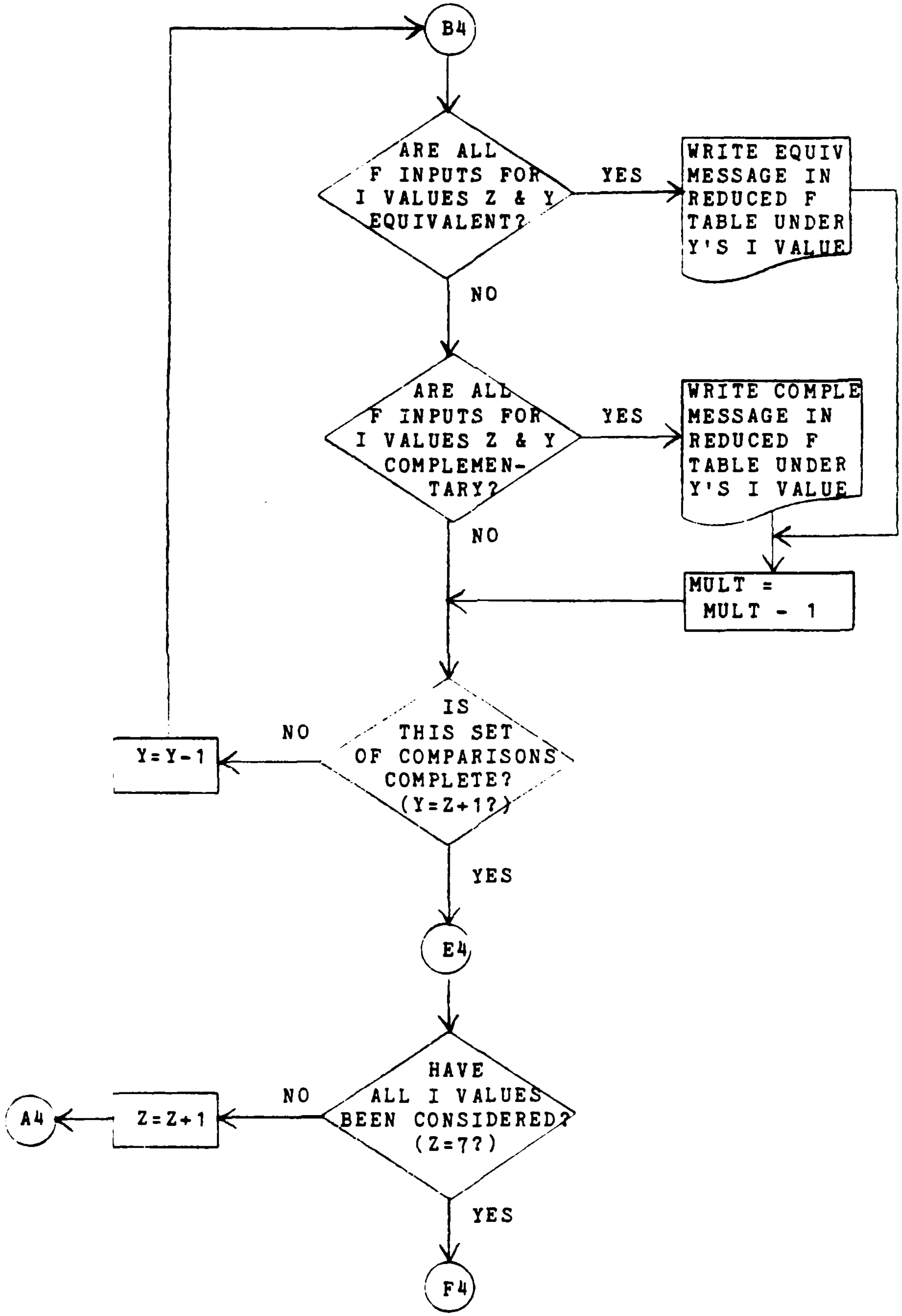

B3
NO MINTERMS FOR THIS I VALUE? (Ictr(X) =0?)
YES
I3
NO
EIGHT MINTERMS FOR THIS I VALUE? (Ictr(X) =8?)
YES
ASSIGN F INPUTS FOR THIS I VALUE TO 1
PTR=PTR+8
I3
NO
C3
MORE THAN ONE MINTERM LEFT FOR THIS I VALUE? (Ictr(X) >1?)
NO
CALCULATE F INDEX
F3
YES
CURRENT MINTERM'S D=1 AND E=1?
YES
F INDEX=3
E3
NO
CURRENT MINTERM'S D=1 AND E=0?
YES
F INDEX=2
E3
NO
D3

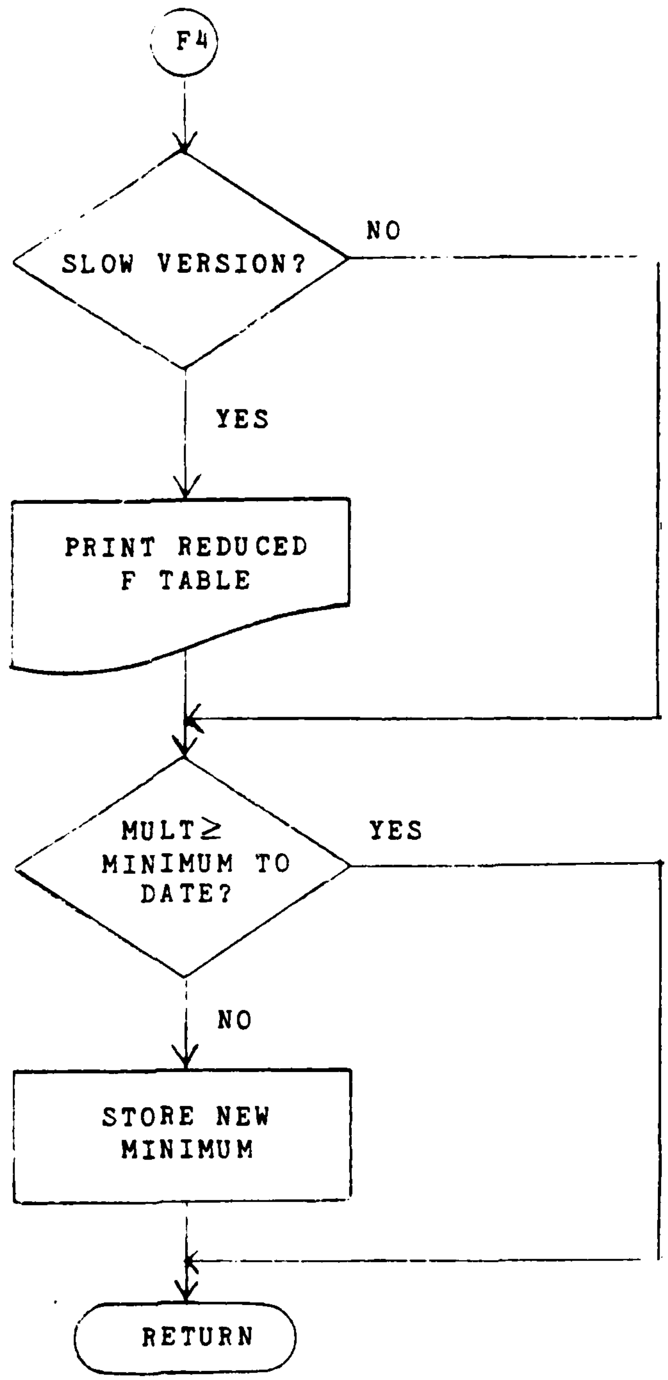

D3
CURRENT MINTERM'S D=0 AND E=1?
YES
F INDEX=1
NO
F INDEX=0
E3
NEXT MINTERM'S D & E VALUES SAME AS CURRENT
NO
YES
FLAG=1
F3
FLAG = 1?
NO
H3
YES
G3


G3
H3
ASSIGN F INPUT FOR THIS F INDEX AND I VALUE TO ONE
ASSIGN F INPUT FOR THIS F INDEX & I VALUE BY MINTERM'S F VALUE
DECREMENT Ictr(x) BY 2
DECREMENT Ictr(x) BY 1
INCREMENT Ptr BY 2
INCREMENT Ptr BY 1
RESET FLAG
ALL TERMS FOR THIS I VALUE CONSIDERED? (Ictr(x) =0?)
NO
C3
YES
I3
HAVE ALL I VALUES BEEN EVALUATED? (X=7?)
NO
X=X+1
B3
YES
J3


J3
SLOW VERSION?
NO
YES
PRINT UNREDUCED F TABLE
RETURN


SUBROUTINE REDUCE
MULT=8
ESTABLISH OUTER I VALUES COUNTER (Z=0)
A4
ARE ALL F INPUTS FOR THIS I VALUE = TO ONE?
YES
WRITE ALL 1'S MESSAGE IN REDUCED F TABLE
MULT = MULT-1
E4
NO
ARE ALL F INPUTS FOR THIS I VALUE = TO ZERO?
YES
WRITE ALL 0'S MESSAGE IN REDUCED F TABLE
MULT = MULT-1
E4
NO
ESTABLISH INNER I VALUES COUNTER FOR COMPARISONS (Y=7)
B4


B4
ARE ALL F INPUTS FOR I VALUES Z & Y EQUIVALENT?
YES
WRITE EQUIV MESSAGE IN REDUCED F TABLE UNDER Y'S I VALUE
NO
ARE ALL F INPUTS FOR I VALUES Z & Y COMPLEMENTARY?
YES
WRITE COMPLE MESSAGE IN REDUCED F TABLE UNDER Y'S I VALUE
NO
MULT = MULT - 1
IS THIS SET OF COMPARISONS COMPLETE? (Y=Z+1?)
NO
Y=Y-1
YES
E4
HAVE ALL I VALUES BEEN CONSIDERED? (Z=7?)
NO
Z=Z+1
A4
YES
F4


F4
SLOW VERSION?
NO
YES
PRINT REDUCED F TABLE
MULT ≥ MINIMUM TO DATE?
YES
NO
STORE NEW MINIMUM
RETURN

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:

COMPLETE DETAILED OUTPUT FOR THE FUNCTION IN EXAMPLE 1

LIST OF MINTERMS

ORIGINAL MINTERM	BINARY EQUIVALENT
3	000011
7	000111
12	001100
14	001110
15	001111
19	010011
23	010111
27	011011
28	011100
29	011101
31	011111
35	100011
39	100111
44	101100
45	101101
46	101110
48	110000
49	110001
50	110010
52	110100
53	110101
55	110111
56	111000
57	111001
59	111011

LIST OF PERMUTATIONS

NUMBER	PERMUTATION	NUMBER	PERMUTATION
1	ABCDEF	11	DEFABC
2	ABDCEF	12	CEFABD
3	ABECDF	13	CDFABE
4	ABFCDE	14	CDEABF
5	ACDBEF	15	BEFACD
6	ACEBDF	16	BDFACE
7	ACFBDE	17	BDEACF
8	ADEBCF	18	BCFADE
9	ADFBCE	19	BCEADF
10	AEFBCD	20	BCDAEF

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ABCDEF				
	000011	000011	000011	0
	000111	000111	000111	0
	001100	001100	001100	1
	001110	001110	001110	1
	001111	001111	001111	1
	010011	010011	010011	2
	010111	010111	010111	2
	011011	011011	011011	3
	011100	011100	011100	3
	011101	011101	011101	3
	011111	011111	011111	3
	100011	100011	100011	4
	100111	100111	100111	4
	101100	101100	101100	5
	101101	101101	101101	5
	101110	101110	101110	5
	110000	110000	110000	6
	110001	110001	110001	6
	110010	110010	110010	6
	110100	110100	110100	6
	110101	110101	110101	6
	110111	110111	110111	6
	111000	111000	111000	7
	111001	111001	111001	7
	111011	111011	111011	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	F	0	F
1	0	0	~F	1
2	0	F	0	F
3	0	F	1	F
4	0	F	0	F
5	0	0	1	~F
6	1	~F	1	F
7	1	F	0	0

REDUCED F MATRIX MULTIPLEXERS=6

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT		TO	I4
1	0	0	~F	1	
2	THIS	EQUIVALENT		TO	I4
3	0	F	1	F	
4	0	F	0	F	
5	0	0	1	~F	
6	1	~F	1	F	
7	1	F	0	0	

THIS IS A NEW MINIMUM: ABCDEF

MULTIPLEXERS = 6

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ABDCEF				
	000011	000011	000011	0
	000111	001011	001011	1
	001100	001100	001100	1
	001110	001110	001110	1
	001111	001111	001111	1
	010011	010011	010011	2
	010111	011011	010111	2
	011011	010111	011011	3
	011100	011100	011100	3
	011101	011101	011101	3
	011111	011111	011111	3
	100011	100011	100011	4
	100111	101011	101011	5
	101100	101100	101100	5
	101101	101101	101101	5
	101110	101110	101110	5
	110000	110000	110000	6
	110001	110001	110001	6
	110010	110010	110010	6
	110100	111000	110100	6
	110101	111001	110101	6
	110111	111011	110111	6
	111000	110100	111000	7
	111001	110101	111001	7
	111011	110111	111011	7

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	0
1	0	F	~F	1
2	0	F	0	F
3	0	F	1	F
4	0	F	0	0
5	0	F	1	~F
6	1	~F	1	F
7	1	F	0	0

REDUCED F MATRIX MULTIPLEXERS=7

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	THIS EQUIVALENT TO I4			
1	0	F	~F	1
2	0	F	0	F
3	0	F	1	F
4	0	F	0	0
5	0	F	1	~F
6	1	~F	1	F
7	1	F	0	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ABECDF				
	000011	001001	000110	0
	000111	001011	001001	1
	001100	000110	001011	1
	001110	001110	001110	1
	001111	001111	001111	1
	010011	011001	010110	2
	010111	011011	010111	2
	011011	011101	011001	3
	011100	010110	011011	3
	011101	010111	011101	3
	011111	011111	011111	3
	100011	101001	100110	4
	100111	101011	100111	4
	101100	100110	101001	5
	101101	100111	101011	5
	101110	101110	101110	5
	110000	110000	110000	6
	110001	110001	110001	6
	110010	111000	110010	6
	110100	110010	110011	6
	110101	110011	110100	6
	110111	111011	110101	6
	111000	110100	111000	7
	111001	110101	111011	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE			
	0	1	2	3
0	0	0	0	~F
1	F	F	0	1
2	0	0	0	1
3	F	F	F	F
4	0	0	0	1
5	F	F	0	~F
6	1	1	1	0
7	~F	F	F	0

REDUCED F MATRIX MULTIPLEXERS=6

I VALUE	g2a(D,E) VALUE				
	0	1	2	3	
0	0	0	0	~F	
1	F	F	0	1	
2	THIS	COMPLEMENT	OF		I6
3	F	F	F	F	
4	THIS	COMPLEMENT	OF		I6
5	F	F	0	~F	
6	1	1	1	0	
7	~F	F	F	0	

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ABFCDE				
	000011	001001	000110	0
	000111	001011	000111	0
	001100	000110	001001	1
	001110	000111	001011	1
	001111	001111	001111	1
	010011	011001	010110	2
	010111	011011	011001	3
	011011	011101	011011	3
	011100	010110	011101	3
	011101	011110	011110	3
	011111	011111	011111	3
	100011	101001	100110	4
	100111	101011	100111	4
	101100	100110	101001	5
	101101	101110	101011	5
	101110	100111	101110	5
	110000	110000	110000	6
	110001	111000	110001	6
	110010	110001	110010	6
	110100	110010	110100	6
	110101	111010	111000	7
	110111	111011	111010	7
	111000	110100	111011	7
	111001	111100	111100	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	1
1	F	F	0	F
2	0	0	0	~F
3	F	F	F	1
4	0	0	0	1
5	F	F	0	~F
6	1	~F	~F	0
7	~F	1	1	0

REDUCED F MATRIX MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT		TO	I4
1	F	F	0	F	
2	0	0	0	~F	
3	F	F	F	1	
4	0	0	0	1	
5	F	F	0	~F	
6	1	~F	~F	0	
7	~F	1	1	0	

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ACDBEF				
	000011	000011	000011	0
	000111	001011	000111	0
	001100	011000	001011	1
	001110	011010	001111	1
	001111	011011	010111	2
	010011	000111	011000	3
	010111	001111	011010	3
	011011	010111	011011	3
	011100	011100	011100	3
	011101	011101	011101	3
	011111	011111	011111	3
	100011	100011	100011	4
	100111	101011	100100	4
	101100	111000	100101	4
	101101	111001	100110	4
	101110	111010	101011	5
	110000	100100	101100	5
	110001	100101	101101	5
	110010	100110	101111	5
	110100	101100	110100	6
	110101	101101	110101	6
	110111	101111	110111	6
	111000	110100	111000	7
	111001	110101	111001	7
	111011	110111	111010	7

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	F
1	0	F	0	F
2	0	0	0	F
3	~F	1	1	F
4	0	F	1	~F
5	0	F	1	F
6	0	0	1	F
7	1	~F	0	0

REDUCED F MATRIX

MULTIPLEXERS=7

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	THIS EQUIVALENT TO I1			
1	0	F	0	F
2	0	0	0	F
3	~F	1	1	F
4	0	F	1	~F
5	0	F	1	F
6	0	0	1	F
7	1	~F	0	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ACEBDF				
	000011	001001	001001	1
	000111	001011	001011	1
	001100	010010	001101	1
	001110	011010	001111	1
	001111	011011	010010	2
	010011	001101	010110	2
	010111	001111	010111	2
	011011	011101	011010	3
	011100	010110	011011	3
	011101	010111	011101	3
	011111	011111	011111	3
	100011	101001	100100	4
	100111	101011	100101	4
	101100	110010	100110	4
	101101	110011	100111	4
	101110	111010	101001	5
	110000	100100	101011	5
	110001	100101	101100	5
	110010	101100	101111	5
	110100	100110	110010	6
	110101	100111	110011	6
	110111	101111	110100	6
	111000	110100	110101	6
	111001	110101	111010	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	F	F	F
2	0	~F	0	1
3	0	1	F	F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	0	~F	F	0

REDUCED F MATRIX MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL	VALUES	=	ZERO
1	F	F	F	F
2	0	~F	0	1
3	0	1	F	F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	0	~F	F	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ACFBDE				
	000011	001001	001001	1
	000111	001011	001011	1
	001100	010010	001101	1
	001110	010011	001111	1
	001111	011011	010010	2
	010011	001101	010011	2
	010111	001111	010110	2
	011011	011101	011011	3
	011100	010110	011101	3
	011101	011110	011110	3
	011111	011111	011111	3
	100011	101001	100100	4
	100111	101011	100101	4
	101100	110010	100110	4
	101101	111010	101001	5
	101110	110011	101011	5
	110000	100100	101100	5
	110001	101100	101110	5
	110010	100101	101111	5
	110100	100110	110010	6
	110101	101110	110011	6
	110111	101111	110100	6
	111000	110100	111010	7
	111001	111100	111100	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	F	F	F
2	0	1	0	¬F
3	0	F	F	1
4	0	0	1	¬F
5	F	F	¬F	1
6	0	1	¬F	0
7	0	¬F	1	0

REDUCED F MATRIX MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL VALUES = ZERO			
1	F	F	F	F
2	0	1	0	¬F
3	0	F	F	1
4	0	0	1	¬F
5	F	F	¬F	1
6	0	1	¬F	0
7	0	¬F	1	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ADEBCF				
	000011	001001	001001	1
	000111	011001	001101	1
	001100	010010	001111	1
	001110	011010	010010	2
	001111	011011	010110	2
	010011	001101	010111	2
	010111	011101	011001	3
	011011	001111	011010	3
	011100	010110	011011	3
	011101	010111	011101	3
	011111	011111	011111	3
	100011	101001	100100	4
	100111	111001	100101	4
	101100	110010	100110	4
	101101	110011	100111	4
	101110	111010	101001	5
	110000	100100	101100	5
	110001	100101	101111	5
	110010	101100	110010	6
	110100	110100	110011	6
	110101	110101	110100	6
	110111	111101	110101	6
	111000	100110	111001	7
	111001	100111	111010	7
	111011	101111	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	0	F	F
2	0	$\bar{F}$	0	1
3	F	1	F	F
4	0	0	1	1
5	F	0	$\bar{F}$	F
6	0	1	1	0
7	F	$\bar{F}$	F	0

REDUCED F MATRIX MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL VALUES = ZERO			
1	F	0	F	F
2	0	$\bar{F}$	0	1
3	F	1	F	F
4	0	0	1	1
5	F	0	$\bar{F}$	F
6	0	1	1	0
7	F	$\bar{F}$	F	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
ADFBCE				
	000011	001001	001001	1
	000111	011001	001101	1
	001100	010010	001111	1
	001110	010011	010010	2
	001111	011011	010011	2
	010011	001101	010110	2
	010111	011101	011001	3
	011011	001111	011011	3
	011100	010110	011101	3
	011101	011110	011110	3
	011111	011111	011111	3
	100011	101001	100100	4
	100111	111001	100101	4
	101100	110010	100110	4
	101101	111010	101001	5
	101110	110011	101100	5
	110000	100100	101110	5
	110001	101100	101111	5
	110010	100101	110010	6
	110100	110100	110011	6
	110101	111100	110100	6
	110111	111101	111001	7
	111000	100110	111010	7
	111001	101110	111100	7
	111011	101111	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	0	F	F
2	0	1	0	¬F
3	F	F	F	1
4	0	0	1	¬F
5	F	0	¬F	1
6	0	1	¬F	0
7	F	¬F	1	0

REDUCED F MATRIX

MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL	VALUES	=	ZERO
1	F	0	F	F
2	0	1	0	¬F
3	F	F	F	1
4	0	0	1	¬F
5	F	0	¬F	1
6	0	1	¬F	0
7	F	¬F	1	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
AEFBCD				
	000011	011000	000011	0
	000111	011001	000111	0
	001100	000011	001111	1
	001110	010011	010011	2
	001111	011011	011000	3
	010011	011100	011001	3
	010111	011101	011011	3
	011011	011110	011100	3
	011100	000111	011101	3
	011101	001111	011110	3
	011111	011111	011111	3
	100011	111000	100011	4
	100111	111001	100100	4
	101100	100011	100101	4
	101101	101011	100110	4
	101110	110011	101011	5
	110000	100100	101100	5
	110001	101100	101101	5
	110010	110100	101110	5
	110100	100101	110011	6
	110101	101101	110100	6
	110111	111101	111000	7
	111000	100110	111001	7
	111001	101110	111101	7
	111011	111110	111110	7

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	F
1	0	0	0	F
2	0	F	0	0
3	1	F	1	1
4	0	F	1	~F
5	0	F	1	~F
6	0	F	~F	0
7	1	0	F	~F

REDUCED F MATRIX

MULTIPLEXERS=7

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	F
1	0	0	0	F
2	0	F	0	0
3	1	F	1	1
4	THIS EQUIVALENT TO I5			
5	0	F	1	~F
6	0	F	~F	0
7	1	0	F	~F

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
DEFABC				
	000011	011000	000110	0
	000111	111000	000111	0
	001100	100001	001110	1
	001110	110001	001111	1
	001111	111001	010110	2
	010011	011010	011000	3
	010111	111010	011010	3
	011011	011011	011011	3
	011100	100011	011100	3
	011101	101011	011111	3
	011111	111011	100001	4
	100011	011100	100011	4
	100111	111100	100101	4
	101100	100101	100110	4
	101101	101101	101011	5
	101110	110101	101101	5
	110000	000110	101110	5
	110001	001110	110001	6
	110010	010110	110101	6
	110100	100110	111000	7
	110101	101110	111001	7
	110111	111110	111010	7
	111000	000111	111011	7
	111001	001111	111100	7
	111011	011111	111110	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	1
1	0	0	0	1
2	0	0	0	~F
3	~F	1	~F	F
4	F	F	F	~F
5	0	F	F	~F
6	F	0	F	0
7	1	1	~F	~F

REDUCED F MATRIX

MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	THIS EQUIVALENT TO I1			
1	0	0	0	1
2	0	0	0	~F
3	~F	1	~F	F
4	F	F	F	~F
5	0	F	F	~F
6	F	0	F	0
7	1	1	~F	~F

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
CEFABD				
	000011	011000	000110	0
	000111	011001	000111	0
	001100	100001	001110	1
	001110	110001	001111	1
	001111	111001	010110	2
	010011	011010	011000	3
	010111	011011	011001	3
	011011	111010	011010	3
	011100	100011	011011	3
	011101	101011	011100	3
	011111	111011	011101	3
	100011	011100	011111	3
	100111	011101	100001	4
	101100	100101	100011	4
	101101	101101	100101	4
	101110	110101	100110	4
	110000	000110	101011	5
	110001	001110	101101	5
	110010	010110	101110	5
	110100	000111	110001	6
	110101	001111	110101	6
	110111	011111	111001	7
	111000	100110	111010	7
	111001	101110	111011	7
	111011	111110	111110	7

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	0	0	1
1	0	0	0	1
2	0	0	0	~F
3	1	1	1	F
4	F	F	F	~F
5	0	F	F	~F
6	F	0	F	0
7	F	1	0	~F

REDUCED F MATRIX MULTIPLEXERS=6

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	THIS EQUIVALENT TO I1			
1	0	0	0	1
2	THIS COMPLEMENT OF I3			
3	1	1	1	F
4	F	F	F	~F
5	0	F	F	~F
6	F	0	F	0
7	F	1	0	~F

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
CDFABE				
	000011	001001	000110	0
	000111	011001	000111	0
	001100	110000	001001	1
	001110	110001	001011	1
	001111	111001	001101	1
	010011	001011	001110	1
	010111	011011	010110	2
	011011	101011	011001	3
	011100	110010	011011	3
	011101	111010	011101	3
	011111	111011	011110	3
	100011	001101	011111	3
	100111	011101	100110	4
	101100	110100	101011	5
	101101	111100	101110	5
	101110	110101	101111	5
	110000	000110	110000	6
	110001	001110	110001	6
	110010	000111	110010	6
	110100	010110	110100	6
	110101	011110	110101	6
	110111	011111	111001	7
	111000	100110	111010	7
	111001	101110	111011	7
	111011	101111	111100	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	1
1	F	F	F	~F
2	0	0	0	~F
3	F	F	F	1
4	0	0	0	~F
5	0	F	0	1
6	1	~F	1	0
7	F	1	~F	0

REDUCED F MATRIX MULTIPLEXERS=6

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	0	0	0	1	
1	F	F	F	~F	
2	THIS	EQUIVALENT		TO	I4
3	F	F	F	1	
4	0	0	0	~F	
5	THIS	COMPLEMENT		OF	I6
6	1	~F	1	0	
7	F	1	~F	0	

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
CDEABF				
	000011	001001	000110	0
	000111	011001	000111	0
	001100	110000	001001	1
	001110	111000	001011	1
	001111	111001	001101	1
	010011	001011	001110	1
	010111	011011	010110	2
	011011	101011	010111	2
	011100	110010	011001	3
	011101	110011	011011	3
	011111	111011	011101	3
	100011	001101	011111	3
	100111	011101	100110	4
	101100	110100	100111	4
	101101	110101	101011	5
	101110	111100	101111	5
	110000	000110	110000	6
	110001	000111	110010	6
	110010	001110	110011	6
	110100	010110	110100	6
	110101	010111	110101	6
	110111	011111	111000	7
	111000	100110	111001	7
	111001	100111	111011	7
	111011	101111	111100	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	1
1	F	F	F	~F
2	0	0	0	1
3	F	F	F	F
4	0	0	0	1
5	0	F	0	F
6	~F	1	1	0
7	1	F	~F	0

REDUCED F MATRIX MULTIPLEXERS=6

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT		TO	I4
1	F	F	F	~F	
2	THIS	EQUIVALENT		TO	I4
3	F	F	F	F	
4	0	0	0	1	
5	0	F	0	F	
6	~F	1	1	0	
7	1	F	~F	0	

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BEFACD				
	000011	011000	000011	0
	000111	011001	000111	0
	001100	000011	001111	1
	001110	010011	010011	2
	001111	011011	010111	2
	010011	111000	011000	3
	010111	111001	011001	3
	011011	111010	011011	3
	011100	100011	011100	3
	011101	101011	011101	3
	011111	111011	100011	4
	100011	011100	100100	4
	100111	011101	100101	4
	101100	000111	100110	4
	101101	001111	101011	5
	101110	010111	101100	5
	110000	100100	101101	5
	110001	101100	101110	5
	110010	110100	110100	6
	110100	100101	111000	7
	110101	101101	111001	7
	110111	111101	111010	7
	111000	100110	111011	7
	111001	101110	111101	7
	111011	111110	111110	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	F	0	F
1	0	0	0	F
2	0	F	0	F
3	1	F	1	0
4	0	F	1	~F
5	0	F	1	~F
6	0	0	~F	0
7	1	1	F	~F

REDUCED F MATRIX MULTIPLEXERS=6

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT		TO	I2
1	0	0	0	F	
2	0	F	0	F	
3	1	F	1	0	
4	THIS	EQUIVALENT		TO	I5
5	0	F	1	~F	
6	0	0	~F	0	
7	1	1	F	~F	

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BDFACE				
	000011	001001	001001	1
	000111	011001	001101	1
	001100	010010	010010	2
	001110	010011	010011	2
	001111	011011	010110	2
	010011	101001	010111	2
	010111	111001	011001	3
	011011	101011	011011	3
	011100	110010	011101	3
	011101	111010	011110	3
	011111	111011	100100	4
	100011	001101	100101	4
	100111	011101	100110	4
	101100	010110	101001	5
	101101	011110	101011	5
	101110	010111	101100	5
	110000	100100	101110	5
	110001	101100	101111	5
	110010	100101	110010	6
	110100	110100	110100	6
	110101	111100	111001	7
	110111	111101	111010	7
	111000	100110	111011	7
	111001	101110	111100	7
	111011	101111	111101	7

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	0	0	0
1	F	0	F	0
2	0	1	0	1
3	F	F	F	~F
4	0	0	1	~F
5	F	F	~F	1
6	0	~F	~F	0
7	F	1	1	0

REDUCED F MATRIX

MULTIPLEXERS=7

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	ALL VALUES = ZERO			
1	F	0	F	0
2	0	1	0	1
3	F	F	F	~F
4	0	0	1	~F
5	F	F	~F	1
6	0	~F	~F	0
7	F	1	1	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BDEACF				
	000011	001001	001001	1
	000111	011001	001101	1
	001100	010010	010010	2
	001110	011010	010110	2
	001111	011011	010111	2
	010011	101001	011001	3
	010111	111001	011010	3
	011011	101011	011011	3
	011100	110010	011101	3
	011101	110011	011110	3
	011111	111011	100100	4
	100011	001101	100101	4
	100111	011101	100110	4
	101100	010110	100111	4
	101101	010111	101001	5
	101110	011110	101011	5
	110000	100100	101100	5
	110001	100101	101111	5
	110010	101100	110010	6
	110100	110100	110011	6
	110101	110101	110100	6
	110111	111101	110101	6
	111000	100110	111001	7
	111001	100111	111011	7
	111011	101111	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	0	F	0
2	0	~F	0	1
3	F	1	F	~F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	F	F	F	0

REDUCED F MATRIX

MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL	VALUES	=	ZERO
1	F	0	F	0
2	0	~F	0	1
3	F	1	F	~F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	F	F	F	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BCFADE				
	000011	001001	001001	1
	000111	001011	001011	1
	001100	010010	001101	1
	001110	010011	001111	1
	001111	011011	010010	2
	010011	101001	010011	2
	010111	101011	010110	2
	011011	111001	010111	2
	011100	110010	011011	3
	011101	111010	011110	3
	011111	111011	100100	4
	100011	001101	100101	4
	100111	001111	100110	4
	101100	010110	101001	5
	101101	011110	101011	5
	101110	010111	101100	5
	110000	100100	101110	5
	110001	101100	101111	5
	110010	100101	110010	6
	110100	100110	110100	6
	110101	101110	111001	7
	110111	101111	111010	7
	111000	110100	111011	7
	111001	111100	111100	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	F	F	F
2	0	1	0	1
3	0	F	0	¬F
4	0	0	1	¬F
5	F	F	¬F	1
6	0	¬F	¬F	0
7	F	1	1	0

REDUCED F MATRIX MULTIPLEXERS=7

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL	VALUES	=	ZERO
1	F	F	F	F
2	0	1	0	1
3	0	F	0	¬F
4	0	0	1	¬F
5	F	F	¬F	1
6	0	¬F	¬F	0
7	F	1	1	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BCEADF				
	000011	001001	001001	1
	000111	001011	001011	1
	001100	010010	001101	1
	001110	011010	001111	1
	001111	011011	010010	2
	010011	101001	010110	2
	010111	101011	010111	2
	011011	111001	011010	3
	011100	110010	011011	3
	011101	110011	011110	3
	011111	111011	100100	4
	100011	001101	100101	4
	100111	001111	100110	4
	101100	010110	100111	4
	101101	010111	101001	5
	101110	011110	101011	5
	110000	100100	101100	5
	110001	100101	101111	5
	110010	101100	110010	6
	110100	100110	110011	6
	110101	100111	110100	6
	110111	101111	110101	6
	111000	110100	111001	7
	111001	110101	111011	7
	111011	111101	111101	7

UNREDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3
0	0	0	0	0
1	F	F	F	F
2	0	~F	0	1
3	0	1	0	~F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	F	F	F	0

REDUCED F MATRIX

MULTIPLEXERS=7

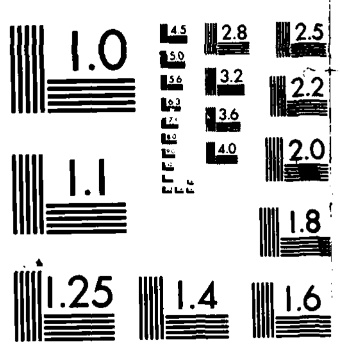
I VALUE	g2a(D,E) VALUE 0	1	2	3
0	ALL	VALUES	=	ZERO
1	F	F	F	F
2	0	~F	0	1
3	0	1	0	~F
4	0	0	1	1
5	F	F	~F	F
6	0	1	1	0
7	F	F	F	0

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BCDAEF				
	000011	000011	000011	0
	000111	001011	000111	0
	001100	011000	001011	1
	001110	011010	001111	1
	001111	011011	011000	3
	010011	100011	011010	3
	010111	101011	011011	3
	011011	110011	011100	3
	011100	111000	011101	3
	011101	111001	011110	3
	011111	111011	100011	4
	100011	000111	100100	4
	100111	001111	100101	4
	101100	011100	100110	4
	101101	011101	101011	5
	101110	011110	101100	5
	110000	100100	101101	5
	110001	100101	101111	5
	110010	100110	110011	6
	110100	101100	110100	6
	110101	101101	110101	6
	110111	101111	110111	6
	111000	110100	111000	7
	111001	110101	111001	7
	111011	110111	111011	7

AD-A161 718 MINIMIZATION OF A SIX VARIABLE BOOLEAN FUNCTION(U) 2/2
NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA SYSTEMS
DIRECTORATE L M KOCH MAY 85 NADC-85135-20

UNCLASSIFIED F/G 9/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

UNREDUCED F MATRIX

	g2a(D,E) VALUE			
I VALUE	0	1	2	3
0	0	F	0	F
1	0	F	0	F
2	0	0	0	0
3	¬F	1	1	¬F
4	0	F	1	¬F
5	0	F	1	F
6	0	F	1	F
7	1	F	0	0

REDUCED F MATRIX   MULTIPLEXERS=5

	g2a(D,E) VALUE				
I VALUE	0	1	2	3	
0	THIS	EQUIVALENT		TO	I1
1	0	F	0	F	
2	ALL	VALUES	=	ZERO	
3	¬F	1	1	¬F	
4	0	F	1	¬F	
5	THIS	EQUIVALENT		TO	I6
6	0	F	1	F	
7	1	F	0	0	

THIS IS A NEW MINIMUM: BCDAEF

MULTIPLEXERS =   5

MINIMUM MULTIPLEXERS NEEDED IS 5

IT IS PERMUTATION NUMBER 20 WHICH IS THE FOLLOWING:

PERMUTATION	ORIGINAL MINTERM	MINTERMS AFTER PERMUTATION	ORDERED MINTERMS	I VALUE
BCDAEF				
	000011	000011	000011	0
	000111	001011	000111	0
	001100	011000	001011	1
	001110	011010	001111	1
	001111	011011	011000	3
	010011	100011	011010	3
	010111	101011	011011	3
	011011	110011	011100	3
	011100	111000	011101	3
	011101	111001	011110	3
	011111	111011	100011	4
	100011	000111	100100	4
	100111	001111	100101	4
	101100	011100	100110	4
	101101	011101	101011	5
	101110	011110	101100	5
	110000	100100	101101	5
	110001	100101	101111	5
	110010	100110	110011	6
	110100	101100	110100	6
	110101	101101	110101	6
	110111	101111	110111	6
	111000	110100	111000	7
	111001	110101	111001	7
	111011	110111	111011	7

THE INPUTS INTO THESE MULTIPLEXERS ARE AS FOLLOWS:

REDUCED F MATRIX

I VALUE	g2a(D,E) VALUE 0	1	2	3	
0	THIS	EQUIVALENT	TO		11
1	0	F	0	F	
2	ALL	VALUES	=	ZERO	
3	¬F	1	1	¬F	
4	0	F	1	¬F	
5	THIS	EQUIVALENT	TO		I6
6	0	F	1	F	
7	1	F	0	0	

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C:

MINIZE PROGRAM LISTING

```
10  ! **************************************************************************
20  !
30  !                    PROGRAM   MINIZE
40  !
50  !
60  !        THIS PROGRAM WILL REALIZE A SIX VARIABLE FUNCTION IN A
70  !        MINIMUM NUMBER OF 4-INPUT MULTIPLEXERS FOR A COMBINATIONAL
80  !        CIRCUIT.
90  !
100 !        WRITTEN BY:  LORRAINE M. KOCH
110 !                     NAVAL AIR DEVELOPMENT CENTER
120 !                     WARMINSTER, PA  18974
130 !                     (215) 441-1595
140 !
150 !
160 !              DATE:  DECEMBER 1984
170 !
180 ! **************************************************************************
190 !
200  OPTION BASE 0
210  COM Orig_ele(63),Orig_array(63,5),Bin_equiv(63,5),Number
220  COM Min_mult,Min_i_inp$(7,3),Mult,Min_where
230  COM Perm(19,5),Alpha_perm$(19,5)
240  COM Work_array(63,5),F$(7,3),I(63),Ictr(7),Point(5),Tempf$(63)
250  COM Zero$,One$,Defalt$,Inp_f$,Inp_nf$
260  COM All1$,All0$,Equiv$,Comple$,Numb$(7),Prt_opt$,Print_option
270  COM Temp_array(63,5),Dividend(5)
280  COM Min_work(63,5),Min_temp(63,5),Min_i(63)
290  INTEGER X,Y,P
300 !
310 ! **************************************************************************
320 !
330 !   CALL TO INTRO SUBROUTINE INTRODUCES THE USER TO THIS PROGRAM
340 !
350 ! **************************************************************************
360 !
370  CALL Intro
380 !
390 ! **************************************************************************
400 !
410 !   CALL TO READ_DATA SUBROUTINE INITIALIZES CERTAIN VARIABLES AND
420 !             SOME DATA ARRAYS.
430 !
440 ! **************************************************************************
450 !
460  CALL Read_data
470 !
480 ! **************************************************************************
490 !
500 !   CALL TO GET_INPUTS PROMPTS THE USER WITH QUESTIONS ON ALL THE
510 !   NECESSARY INPUTS AND OPTIONS AVAILABLE.
520 !
530 ! **************************************************************************
540 !
```

```
550 Inputs:   !
560   CALL Get_inputs
570 !
580  ! *************************************************************************
590  !
600  !            THIS SECTION IS THE HEART OF THIS PROGRAM
610  !
620  ! INDEX P CONTROLS THE PROCESS THROUGH ALL 20 PERMUTATIONS.
630  !
640  !
650  ! ASSIGN_F_VALUES - ASSIGNS THE F INPUTS FOR EACH g2a(D,E) INDEX VALUE
660  !                   UNDER EACH I VALUE FOR THE PERMUTATION THAT IS PASSED.
670  !
680  ! REDUCE          - EVALUATES THE F INPUTS MATRIX AND TRIES TO REDUCE THE
690  !                   NUMBER OF MULTIPLEXERS NEEDED TO REALIZE THIS CASE.
700  !                   THIS ROUTINE ALSO CHECKS AND STORES THE MINUMUM TO DATE
710  !
720  !
730  ! P = THE PERMUTATION NUMBER WE ARE CURRENTLY WORKING ON
740  !
750  ! *************************************************************************
760  !
770   FOR P=0 TO 19
780          CALL Assign_f_values(P)
790          CALL Reduce(P)
800   NEXT P
810  !
820  ! *************************************************************************
830  !
840  !                      WE ARE FINISHED
850  !
860  !    NOW WE WILL PRINTOUT THE MINIMUM NUMBER OF MULTIPLEXERS
870  !    NEEDED TO REALIZE THIS FUNCTION AND ALSO THEIR REQUIRED
880  !    INPUTS.
890  !
900  ! *************************************************************************
910  !
920   PRINTER IS 0
930   BEEP
940   PRINT PAGE
950   PRINT LIN(3)
960   PRINT "          MINIMUM MULTIPLEXERS NEEDED IS";Min_mult
970   PRINT
980   PRINT USING "10X,K,K,2D,K,K";"IT IS PERMU","TATION NUMBER ";Min_where;" WH
ICH IS"," THE FOLLOWING: "
990   Indx=Min_where-1
1000  PRINT
1010 !
1020 ! *************************************************************************
1030 !
1040 ! PRINT THE ORIG , TEMP AND WORK ARRAY OF THE MINIMUM
1050 !
1060 ! *************************************************************************
1070 !
```

```
1080  PRINT "                           ORIGINAL   MINTERMS AFTER   ORDERED
"
1090  PRINT "          PERMUTATION   MINTERM     PERMUTATION     MINTERMS
I VALUE"
1100  PRINT USING "13X,6(K)";Alpha_perm$(Indx,5),Alpha_perm$(Indx,4),Alpha_perm$
(Indx,3),Alpha_perm$(Indx,2),Alpha_perm$(Indx,1),Alpha_perm$(Indx,0)
1120  FOR X=0 TO Number
1130      PRINT USING "#,K";"                          "
1140      FOR Y=5 TO 0 STEP -1
1150          PRINT USING "#,D";Orig_array(X,Y)
1160      NEXT Y
1170      PRINT USING "#,K";"        "
1180      FOR Y=5 TO 0 STEP -1
1190          PRINT USING "#,D";Min_temp(X,Y)
1200      NEXT Y
1210      PRINT USING "#,K";"        "
1220      FOR Y=5 TO 0 STEP -1
1230          PRINT USING "#,D";Min_work(X,Y)
1240      NEXT Y
1250      PRINT USING "8X,D";Min_i(X)
1260  NEXT X
1270  PRINT LIN(2)
1280  IF Number>=30 THEN PRINT PAGE
1290  PRINT "         THE INPUTS INTO THESE MULTIPLEXERS ARE AS FOLLOWS:"
1300  PRINT
1310  PRINT "                        REDUCED F MATRIX"
1320  PRINT
1330  PRINT "                                  g2s(D,E) VALUE    "
1340  PRINT "             I VALUE       0    1    2    3   "
1350  FOR X=0 TO 7
1360      PRINT USING "17X,D,8X,K,K,K,K";X,Min_i_inp$(X,0),Min_i_inp$(X,1),Min
_i_inp$(X,2),Min_i_inp$(X,3)
1370  NEXT X
1380  !
1390  ! ************************************************************************
1400  !
1410  ! DOES THE USER WISH TO EVALUATE ANOTHER FUNCTION.  IF SO, GO BACK TO THE
1420  ! THE GET_INPUTS SUBROUTINE AND BEGIN AGAIN.
1430  !
1440  ! ************************************************************************
1450  !
1460  PRINTER IS 16
1470  PRINT PAGE
1480  Rsvp$="N"
1490  INPUT "Q-6.  DO YOU WISH TO EVALUATE ANOTHER FUNCTION (  Y  OR  N  )?"
,Rsvp$
1500  IF Rsvp$<>"Y" THEN GOTO 1560
1510  Again$="Y"
1511  Min_mult=9
1512  Min_where=0
1520  GOTO Inputs
1530  !
1540  !
1550  BEEP
1560  PRINT " PROGRAM END "
1570  STOP
1580  !
```

```
1590 !
1600 ! ************************************************************************
1610 !
1620 !                         INTRO SUBROUTINE
1630 !
1640 !           INTRO SUBROUTINE INTRODUCES THE USER TO MINIZE
1650 !
1660 ! ************************************************************************
1670 !
1680  SUB Intro
1690  !
1700  PRINTER IS 16
1710  PRINT
1720  PRINT "                            M I N I Z E   "
1730  PRINT LIN(2)
1740  PRINT "        THIS PROGRAM WILL REALIZE A SIX VARIABLE COMBINATIONAL
"
1750  PRINT "    BOOLEAN FUNCTION  F(A,B,C,D,E,F) IN A MINIMUM NUMBER OF "
1760  PRINT "    4-INPUT MULTIPLEXERS.  YOU WILL BE ASKED FOR THE NUMBER OF"
1770  PRINT "    MINTERMS (OR ELEMENTS) AND THEN WHAT THESE MINTERMS ARE."
1780  PRINT
1790  PRINT "         MINIZE WILL OUTPUT TO YOU THE MINIMUM NUMBER OF 4-INPUT"
1800  PRINT "    MULTIPLEXERS YOU WILL NEED AND WHAT EACH OF THE INPUTS ARE"
1810  PRINT "    INTO EACH MULTIPLEXER."
1820  PRINT LIN(5)
1830  PRINT " PRESS  CONT  WHEN YOU ARE READY."
1840  PAUSE
1850  SUBEND
1860 !
1870 ! ************************************************************************
1880 !
1890 !                       READ_DATA SUBROUTINE
1900 !
1910 !  THE READ_DATA SUBROUTINE INITIALIZES CERTAIN DATA ARRAYS AND VARIABLES.
1920 !
1930 !          PERM ARRAY - STORES THE 20 PERMUTATIONS
1940 !      DIVIDEND ARRAY - STORES POWERS OF 2 TO COMPUTE BINARY EQUIVALENTS
1950 !            MIN_MULT - STORES MINIMUM NUMBER OF MULTIPLEXERS REQUIRED
1960 !                       TO DATE
1970 !           MIN_WHERE - STORES PERMUTATION NUMER OF MINIMUM TO DATE
1980 !
1990 !       THE ALPHA-NUMERIC ARRAYS STORE MESSAGES USED IN CALCULATING
2000 !             THE UNREDUCED AND REDUCED F INPUTS MATRICES.
2010 !
2020 ! ************************************************************************
2030 !
2040  SUB Read_data
2050  OPTION BASE 0
2060  COM Orig_ele(*),Orig_array(*),Bin_equiv(*),Number
2070  COM Min_mult,Min_i_inp$(*),Mult,Min_where
2080  COM Perm(*),Alpha_perm$(*)
2090  COM Work_array(*),F$(*),I(*),Ictr(*),Point(*),Tempf$(*)
2100  COM Zero$,One$,Defalt$,Inp_f$,Inp_nf$
2110  COM All1$,All0$,Equiv$,Comple$,Numb$(*),Prt_opt$,Print_option
2120  COM Temp_array(*),Dividend(*)
2130  COM Min_work(*),Min_temp(*),Min_i(*)
```

```
2140  INTEGER Perm,Dividend
2150  !
2160  DATA 0,1,2,3,4,5,        A,B,C,D,E,F
2170  DATA 0,1,3,2,4,5,        A,B,D,C,E,F
2180  DATA 0,2,3,1,4,5,        A,B,E,C,D,F
2190  DATA 1,2,3,0,4,5,        A,B,F,C,D,E
2200  DATA 0,1,4,2,3,5,        A,C,D,B,E,F
2210  DATA 0,2,4,1,3,5,        A,C,E,B,D,F
2220  DATA 1,2,4,0,3,5,        A,C,F,B,D,E
2230  DATA 0,3,4,1,2,5,        A,D,E,B,C,F
2240  DATA 1,3,4,0,2,5,        A,D,F,B,C,E
2250  DATA 2,3,4,0,1,5,        A,E,F,B,C,D
2260  !
2270  !
2280  DATA 3,4,5,0,1,2,        D,E,F,A,B,C
2290  DATA 2,4,5,0,1,3,        C,E,F,A,B,D
2300  DATA 1,4,5,0,2,3,        C,D,F,A,B,E
2310  DATA 0,4,5,1,2,3,        C,D,E,A,B,F
2320  DATA 2,3,5,0,1,4,        B,E,F,A,C,D
2330  DATA 1,3,5,0,2,4,        B,D,F,A,C,E
2340  DATA 0,3,5,1,2,4,        B,D,E,A,C,F
2350  DATA 1,2,5,0,3,4,        B,C,F,A,D,E
2360  DATA 0,2,5,1,3,4,        B,C,E,A,D,F
2370  DATA 0,1,5,2,3,4,        B,C,D,A,E,F
2380  FOR X=0 TO 19
2390       FOR Y1=0 TO 5
2400            READ Perm(X,Y1)
2410       NEXT Y1
2420       FOR Y2=5 TO 0 STEP -1
2430            READ Alpha_perm$(X,Y2)
2440       NEXT Y2
2450  NEXT X
2460 !
2470  DATA 1,2,4,8,16,32
2480  MAT READ Dividend
2490 !
2500  DATA " ALL VALUES = ONE"
2510  DATA " ALL VALUES = ZERO"
2520  DATA "THIS EQUIVALENT TO"
2530  DATA "THIS COMPLEMENT OF"
2540  READ All1$,All0$,Equiv$,Comple$
2550 !
2560  DATA "I0","I1","I2","I3","I4","I5","I6","I7"
2570  READ Numb$(0),Numb$(1),Numb$(2),Numb$(3),Numb$(4),Numb$(5),Numb$(6),Numb$(
7)
2580 !
2590  Again$="N"
2600  Zero$="  0  "
2610  One$="  1  "
2620  Defalt$="     "
2630  Inp_f$="  F  "
2640  Inp_nf$=" ~F  "
2650  Min_mult=9
2660  Min_where=0
2670  SUBEND
2680 !
```

```
2690 ! **************************************************************************
2700 !
2710 !                          GET_INPUTS SUBROUTINE
2720 !
2730 !   USER DEFINED INPUTS:  1) NUMBER = NUMBER OF MINTERMS IN THIS 6 VARIABLE
2740 !                                  FUNCTION ( MIN=1, MAX=64 )
2750 !                         2) ORIG_ELE(63) = THE ACTUAL MINTERMS UP TO A
2760 !                                        MAXIMUM OF 63; MINIMUM OF 0
2770 !
2780 ! **************************************************************************
2790 !
2800  SUB Get_inputs
2810  OPTION BASE 0
2820  COM Orig_ele(*),Orig_array(*),Bin_equiv(*),Number
2830  COM Min_mult,Min_i_inp$(*),Mult,Min_where
2840  COM Perm(*),Alpha_perm$(*)
2850  COM Work_array(*),F$(*),I(*),Ictr(*),Point(*),Tempf$(*)
2860  COM Zero$,One$,Defalt$,Inp_f$,Inp_nf$
2870  COM All1$,All0$,Equiv$,Comple$,Numb$(*),Prt_opt$,Print_option
2880  COM Temp_array(*),Dividend(*)
2890  COM Min_work(*),Min_temp(*),Min_i(*)
2900  INTEGER Orig_ele,Orig_array,Temp,X,Y,Z,Z1
2910  PRINTER IS 16
2920  PRINT PAGE
2930  MAT Orig_ele=ZER
2940  Number=0
2950  INPUT "Q-1.  ENTER THE NUMBER OF MINTERMS IN THIS 6 VARIABLE FUNCTION:",Nu
mber
2960  PRINT LIN(3)
2970  IF (Number<=64) AND (Number>=1) THEN GOTO 3010
2980  BEEP
2990  PRINT "NUMBER OF MINTERMS MUST BE BETWEEN 1 AND 64 INCLUSIVE"
3000  GOTO 2950
3010  Number=Number-1
3020  REDIM Orig_ele(Number)
3030  INPUT "Q-2.  ENTER THE MINTERMS:",Orig_ele(*)
3040  MAT SORT Orig_ele(*)
3050  IF (Orig_ele(0)>=0) AND (Orig_ele(Number)<=63) THEN GOTO 3090
3060  BEEP
3070  PRINT "MINTERMS MUST BE BETWEEN 0 AND 63 INCLUSIVE"
3080  GOTO 3030
3090  FOR X=0 TO Number-1
3100     IF Orig_ele(X)=Orig_ele(X+1) THEN GOTO 3130
3110  NEXT X
3120  GOTO 3160
3130  PRINT " EACH MINTERM MUST BE UNIQUE"
3140  BEEP
3150  GOTO 3030
3160  PRINT LIN(3)
3170  PRINT "LIST OF MINTERMS IS:";Orig_ele(*)
3180  PRINT LIN(3)
3190  Rsvp$="Y"
3200  INPUT "Q-3.  ARE THESE CORRECT ( Y  OR  N )?",Rsvp$
3210  IF Rsvp$<>"Y" THEN GOTO 2920
3220 !
```

```
3230 ! ***********************************************************************
3240 !
3250 !            ASSIGN THESE MINTERMS THEIR BINARY EQUIVALENT
3260 !
3270 !    ORIG_ARRAY(63,5)=CONTAINS BINARY EQUIVALENT OF THE ORIGINAL
3280 !                     MINTERMS
3290 !
3300 ! ***********************************************************************
3310 !
3320 REDIM Orig_array(Number,5)
3330 PRINTER IS 0
3340 PRINT PAGE
3341 PRINT LIN(4)
3342 PRINT "                             LIST OF MINTERMS"
3343 PRINT
3350 PRINT "                    ORIGINAL MINTERM    BINARY EQUIVALENT"
3360 FOR X=0 TO Number
3370 PRINT USING "#,27X,2D,16X";Orig_ele(X)
3380 Temp=Orig_ele(X)
3390      FOR Y=5 TO 0 STEP -1
3400           Orig_array(X,Y)=INT(Temp/Dividend(Y))
3410           Temp=Temp MOD Dividend(Y)
3420           PRINT USING "#,D";Orig_array(X,Y)
3430      NEXT Y
3440 PRINT
3450 NEXT X
3460 PRINTER IS 16
3470 !
3480 ! ***********************************************************************
3490 !
3500 !  THIS SECTION ALLOWS THE USER TO SELECT THE TYPE OF PRINTOUT
3510 !  THAT IS DESIRED.  THE DETAILED PRINTOUT WILL RUN CONSIDERABLY
3520 !  SLOWER.
3530 !
3540 !         PRT_OPT$ -  F=FAST VERSION  OR  S=SLOW DETAILED VERSION
3550 !    PRINT_OPTION$ -  16=PRINTS EVERYTHING ON THE CRT   OR
3560 !                      0=PRINTS EVERYTHING ON THE THERMAL PRINTER
3570 !
3580 ! ***********************************************************************
3590 !
3600 Prt_opt$="F"
3610 Print_option=16
3620 PRINT "Q-4.  DO YOU WANT TO RUN THE FAST VERSION OR THE SLOW DETAILED"
3630 PRINT "          PRINTOUT VERSION ( F=FAST  OR  S=SLOW )?"
3640 INPUT Prt_opt$
3650 IF Prt_opt$="F" THEN GOTO 3710
3660 PRINT LIN(3)
3670 PRINT "Q-4a.  DO YOU WANT THE DETAILED PRINTOUT TO APPEAR ON THE CRT OR"
3680 PRINT "          DO YOU WANT A HARD COPY ( 16=CRT  OR  0=HARD COPY )?"
3690 INPUT Print_option
3700 !
3710 ! ***********************************************************************
3720 !
3730 !   THIS SECTION ASKS THE USER IF THEY WISH TO LIST THE PERMUTATIONS.
3740 !
```

```
3750 !   IF YES, THE 20 PERMUTATIONS ARE PRINTED ON THE DESIRED PRINTOUT
3760 !       OPTION.
3770 !
3780 ! **************************************************************************
3790 !
3800  PRINT LIN(4)
3810  INPUT "Q-5.  DO YOU WANT TO LIST THE PERMUTATIONS ( Y  OR  N )?",Rsvp$
3820  PRINTER IS Print_option
3830  IF Rsvp$<>"Y" THEN GOTO 4070
3840  PRINT LIN(4)
3850  IF Number>40 THEN PRINT PAGE
3851  PRINT "                                   LIST OF PERMUTATIONS"
3852  PRINT
3860   PRINT "               NUMBER     PERMUTATION        NUMBER      PERMU
TATION"
3870   FOR Z=0 TO 9
3880     PRINT USING "#,12X,3D,11X";Z+1
3890     FOR Z1=5 TO 0 STEP -1
3900         PRINT USING "#,K";Alpha_perm$(Z,Z1)
3910     NEXT Z1
3920     PRINT USING "#,11X,3D,11X";Z+11
3930     FOR Z1=5 TO 0 STEP -1
3940         PRINT USING "#,K";Alpha_perm$(Z+10,Z1)
3950     NEXT Z1
3960   PRINT
3970   NEXT Z
3980 !
3990 ! **************************************************************************
4000 !
4010 !   REDIMENSION THE ARRAYS FOR THE CURRENT NUMBER OF MINTERMS
4020 !
4030 ! **************************************************************************
4040 !
4050  REDIM Work_array(Number,5),Min_work(Number,5),Min_temp(Number,5)
4060  REDIM Min_i(Number),Temp_array(Number,5),Tempf$(Number)
4070   SUBEND
4080 !
4090 ! **************************************************************************
4100 !
4110 !                     ASSIGN_F_VALUES SUBROUTINE
4120 !
4130 !
4140 !      THIS SUBROUTINE ASSIGNS THE F INPUTS TO THE F$ MATRIX
4150 !      FOR THE CURRENT PERMUTATION.  THERE IS ONE F INPUT FOR
4160 !      EACH g2a(D,E) INDEX UNDER EACH I VALUE.
4170 !
4180 ! **************************************************************************
4190 !
4200  SUB Assign_f_values(INTEGER Indx1)
4210  OPTION BASE 0
4220  COM Orig_ele(*),Orig_array(*),Bin_equiv(*),Number
4230  COM Min_mult,Min_i_inp$(*),Mult,Min_where
4240  COM Perm(*),Alpha_perm$(*)
4250  COM Work_array(*),F$(*),I(*),Ictr(*),Point(*),Tempf$(*)
4260  COM Zero$,One$,Defalt$,Inp_f$,Inp_nf$
```

```
4270  COM A111$,A110$,Equiv$,Comple$,Numb$(*),Prt_opt$,Print_option
4280  COM Temp_array(*),Dividend(*)
4290  COM Min_work(*),Min_temp(*),Min_i(*)
4300  INTEGER Where,Point,Perm,I,Ictr,Work_array,Orig_array,Temp_array,Ptr
4310  INTEGER Findex,X,Y
4320  PRINTER IS 16
4330 !
4340 ! **********************************************************************
4350 !
4360 !  DISPLAY ON THE CRT THE MINIMUM TO DATE AND THE CURRENT PERMUTATION #
4370 !
4380 ! **********************************************************************
4390 !
4400  Where=Indx1
4410  PRINT CHR$(27)&"m"
4420  PRINT PAGE
4430  PRINT USING "K,DDD,K,K,D";"PERMUTATION ",Min_where," IS THE MINIMUM-","---
MULTIPLEXORS = ",Min_mult
4440  PRINT
4450  PRINT "THE PROGRAM IS CURRENTLY WORKING ON PERMUTATION  ";Where+1
4460  PRINT CHR$(27)&"1"
4470  PRINTER IS Print_option
4480 !
4490 ! **********************************************************************
4500 !
4510 ! ASSIGN POINT ARRAY TO THE CURRENT PERMUTATION AND INITIALIZE ARRAYS
4520 !
4530 ! **********************************************************************
4540 !
4550  FOR X=0 TO 5
4560      Point(X)=Perm(Indx1,X)
4570  NEXT X
4580  FOR X=0 TO 7
4590      FOR Y=0 TO 3
4600         F$(X,Y)=Zero$
4610      NEXT Y
4620  NEXT X
4630  FOR X=0 TO Number
4640      Tempf$(X)=Defalt$
4650  NEXT X
4660  MAT I=ZER
4670  MAT Ictr=ZER
4680 !
4690 ! **********************************************************************
4700 !
4710 !  COPY ORIG_ARRAY TO WORK_ARRAY AND REORDER WORK_ARRAY ACCORDING TO THE
4720 !  CURRENT PERMUTATION.  THEN SORT THE WORK_ARRAY.
4730 !
4740 ! **********************************************************************
4750 !
4760  MAT Work_array=Orig_array
4770  MAT REORDER Work_array BY Point,2
4780  MAT Temp_array=Work_array
4790  MAT SORT Work_array(*,5),(*,4),(*,3),(*,2),(*,1),(*,0)
4800  PRINT LIN(2)
4810 !
```

```
4820 ! ***********************************************************************
4830 !
4840 !  COUNT THE NUMBER OF MINTERMS FOR EACH I VALUE (0 THROUGH 7)
4850 !       THIS IS THE FUNCTION g1=g1(A,B,C).
4860 !
4870 !  ALSO  FILL TEMPF$ ARRAY WITH VALUE OF LAST ELEMENT (F)
4880 !
4890 ! ***********************************************************************
4900 !
4910  FOR X=0 TO Number
4920      I(X)=Work_array(X,5)*4+Work_array(X,4)*2+Work_array(X,3)
4930      Ictr(I(X))=Ictr(I(X))+1
4940      Tempf$(X)=Inp_f$
4950      IF Work_array(X,0)=0 THEN Tempf$(X)=Inp_nf$
4960  NEXT X
4970 !
4980 ! ***********************************************************************
4990 !
5000 !  IF THE DETAILED PRINTOUT HAS BEEN SELECTED, THEN PRINT:
5010 !
5020 !             POINT = THE CURRENT PERMUTATION
5030 !        ORIG_ARRAY = ORIGINAL MINTERMS
5040 !        TEMP_ARRAY = WORKING ARRAY AFTER THIS PERMUTATION
5050 !        WORK_ARRAY = SORTED WORKING ARRAY
5060 !        I VALUE    = VALUE OF THE FUNCTION g1=g1(A,B,C) WHICH IS THE
5070 !                     FIRST THREE VALUES OF THE REORDERED MINTERM
5080 !
5090 ! ***********************************************************************
5100 !
5110  IF Prt_opt$<>"S" THEN GOTO 5330
5120  PRINT PAGE
5121  PRINT LIN(3)
5130  PRINT "                            ORIGINAL   MINTERMS AFTER   ORDERED
        "
5140  PRINT "           PERMUTATION   MINTERM     PERMUTATION     MINTERMS
   I VALUE"
5150  PRINT USING "13X,6(K)";Alpha_perm$(Indx1,5),Alpha_perm$(Indx1,4),Alpha_per
m$(Indx1,3),Alpha_perm$(Indx1,2),Alpha_perm$(Indx1,1),Alpha_perm$(Indx1,0)
5170  FOR X=0 TO Number
5180  PRINT USING "#,K";"                          "
5190      FOR Y=5 TO 0 STEP -1
5200          PRINT USING "#,D";Orig_array(X,Y)
5210      NEXT Y
5220      PRINT USING "#,K";"         "
5230      FOR Y=5 TO 0 STEP -1
5240          PRINT USING "#,D";Temp_array(X,Y)
5250      NEXT Y
5260      PRINT USING "#,K";"         "
5270      FOR Y=5 TO 0 STEP -1
5280          PRINT USING "#,D";Work_array(X,Y)
5290      NEXT Y
5300      PRINT USING "8X,D";I(X)
5310  NEXT X
5320  PRINT
5330 !
5340 ! ***********************************************************************
5350 !
```

```
5360 ! NOW WE WILL CALCULATE THE F$ MATRIX OF INPUTS FOR THIS PERMUTATION.
5370 ! WE WILL BEGIN TO COMPUTE THE F MATRIX BY CHECKING THE ICTR ARRAY
5380 ! AND COMPARING TWO MINTERMS AT A TIME TO DETERMINE THE F INPUT FOR
5390 ! EACH F INDEX UNDER EACH I VALUE.
5400 !
5410 !     DEFINITION OF TERMS
5420 !
5430 !     I VALUE = g1(A,B,C) VALUE
5440 !     F INDEX = g2a(D,E) VALUE
5450 !   F$ MATRIX = CONTAINS THE F INPUTS; ONE FOR EACH F INDEX UNDER EACH
5460 !               I VALUE
5470 ! IF THEIR IS ONLY ONE MINTERM FOR A PARTICULAR F INDEX AND I VALUE,
5480 ! THE F INPUT ASSIGNED IS DEPENDENT OF THE MINTERM'S F ARGUMENT (F VALUE).
5490 !     F INPUT = " F " (VARIABLE INP_F$) IF MINTERM'S F VALUE IS 1 OR
5500 !             = "~F " (VARIABLE INP_NF$) IF MINTERM'S F VALUE IS 0.
5510 !
5520 !     METHOD
5530 !
5540 !  IF ICTR(X)=0 THEN THERE ARE NO MINTERMS WITH THIS I VALUE.
5550 !                    THE F INPUTS FOR ALL INDICES SHOULD BE ASSIGNED TO 0.
5560 !                    F$ MATRIX IS INITIALIZED TO ZERO.
5570 !            =1 THEN THERE IS ONLY ONE MINTERM LEFT FOR THIS I
5580 !                    VALUE SO ASSIGN F INPUT ACCORDING TO THIS MINTERM'S
5590 !                    F VALUE FOR THE MINTERM'S F INDEX FOR THIS I VALUE.
5600 !            =2 THEN THERE ARE TWO MINTERMS FOR THIS I VALUE; MUST
5610 !                    CHECK WHAT F INDEX CURRENT MINTERM IS AND SEE
5620 !                    IF THE NEXT MINTERM'S IS THE SAME.
5630 !          =3-7 THEN MUST CHECK THE NEXT TWO MINTERMS TO SEE IF THE
5640 !                    F INDEX OF CURRENT IS THE SAME AS NEXT
5650 !            =8 THEN ALL MINTERMS FOR THIS I VALUE, SO F INPUTS FOR ALL
5660 !                    INDICES SHOULD BE ASSIGNED 1 FOR THIS I VALUE.
5670 !
5680 !
5690 !    PTR = POINTER INTO THE WORK_ARRAY OF ALL THE MINTERMS REORDERED
5700 !          AND SORTED FOR THIS PERMUTATION.
5710 !      X = CONTROLS THE COUNTER THROUGH ALL THE I VALUES OF THE F$ MATRIX
5720 ! FINDEX = VARIABLE THAT GETS ASSIGNED THE CURRENT MINTERM'S F INDEX VALUE.
5730 !
5740 ! ***********************************************************************
5750 !
5760  Ptr=0
5770  Flag$=Defalt$
5780  FOR X=0 TO 7
5790      IF Ictr(X)=0 THEN Next_x
5800      IF Ictr(X)<8 THEN Contl
5810         ! HERE THERE IS ALL 8 MINTERMS FOR THIS I VALUE; MUST ASSIGN
5820         ! ALL F INDICES = 1
5830         FOR Y=0 TO 3
5840             F$(X,Y)=One$
5850         NEXT Y
5860         Ptr=Ptr+8
5870         GOTO Next_x
5880 Contl:  !
5890      IF Ictr(X)>1 THEN GOTO Ckindx
5900         ! HERE THERE IS ONLY ONE ELEMENT LEFT. WE MUST FILL THE
```

```
5910           ! APPROPRIATE INDEX OF THE F MATRIX WITH THE F VALUE.
5920           Findex=INT(Work_array(Ptr,2)*2+Work_array(Ptr,1))
5930           GOTO Asign
5940 Ckindx:  ! HERE THERE IS AT LEAST 2 MINTERMS WITH THIS I VALUE;
5950           !      FIND THE CURRENT F INDEX WE ARE WORKING ON
5960        IF (Work_array(Ptr,2)=1) AND (Work_array(Ptr,1)=1) THEN GOTO F3
5970        IF Work_array(Ptr,2)=1 THEN GOTO F2
5980        IF Work_array(Ptr,1)=1 THEN GOTO F1
5990 F0:    Findex=0
6000        !  BOTH D AND E VALUES FOR THIS ELEMENT ARE ZERO
6010        ! NOW CHECK IF NEXT MINTERM'S D AND E VALUES ARE ZERO
6020        ! IF YES, FLAG IS SET -  OTHERWISE ASIGN ROUTINE WILL ASSIGN
6030        ! THE F MATRIX THE APPROPRIATE F INPUT ACCORDING TO THE CURRENT
6040        ! MINTERM'S F VALUE (TEMPF$ ARRAY)
6050        IF (Work_array(Ptr+1,2)=0) AND (Work_array(Ptr+1,1)=0) THEN Flag$=One$
6060        GOTO Asign
6070 F1:    Findex=1
6080        ! D VALUE IS 0 , E VALUE IS 1
6090        ! CHECK IF NEXT MINTERM HAS THE SAME D AND E VALUES
6100        IF (Work_array(Ptr+1,2)=0) AND (Work_array(Ptr+1,1)=1) THEN Flag$=One$
6110        GOTO Asign
6120 F2: Findex=2
6130        ! D VALUE IS 1 , E VALUE IS 0
6140        ! CHECK IF NEXT MINTERM HAS THE SAME D AND E VALUES
6150        IF (Work_array(Ptr+1,2)=1) AND (Work_array(Ptr+1,1)=0) THEN Flag$=One$
6160        GOTO Asign
6170 F3: Findex=3
6180        ! D VALUE IS 1 , E VALUE IS 1
6190        ! CHECK IF NEXT MINTERM HAS THE SAME D AND E VALUES
6200        IF (Work_array(Ptr+1,2)=1) AND (Work_array(Ptr+1,1)=1) THEN Flag$=One$
6210 !
6220 ! ***************************************************************************
6230 !
6240 ! THIS ROUTINE WILL ASSIGN THE APPROPRIATE F INPUT INTO THE F MATRIX
6250 ! FOR THE CURRENT F INDEX AND I VALUE.
6260 !
6270 ! ***************************************************************************
6280 !
6290 Asign:  IF Flag$=Defalt$ THEN GOTO Only1
6300    ! HERE THERE ARE TWO MINTERMS WITH THE SAME F INDEX SO ASSIGN
6310    ! F MATRIX = 1 FOR THIS F INDEX AND I VALUE.  FLAG HAS BEEN ASSIGNED THIS
6320    ! VALUE.
6330    ! DECREMENT/INCREMENT COUNTER AND POINTER BY 2
6340    F$(X,Findex)=Flag$
6350    Ictr(X)=Ictr(X)-2
6360    Ptr=Ptr+2
6370    Flag$=Defalt$
6380    GOTO Ckend
6390 Only1: ! HERE THERE IS ONLY ONE MINTERM WITH THIS F INDEX SO ASSIGN
6400          ! F MATRIX THE APPROPRIATE F INPUT BASED ON THE MINTERM'S F VALUE.
6410          ! DECREMENT/INCREMENT COUNTER AND POINTER BY 1
6420    F$(X,Findex)=Tempf$(Ptr)
6430    Ictr(X)=Ictr(X)-1
6440    Ptr=Ptr+1
6450 Ckend: ! CHECK TO SEE IF ANY MORE MINTERMS WITH THIS I VALUE
```

```
6460        ! IF CTR NOT ZERO THEN CONTINUE WITH THIS I VALUE AND
6470        ! CONTINUE FILLING F MATRIX WITH REQUIRED INPUTS.
6480        ! IF CTR = 0 THEN NO MORE MINTERMS FOR THIS I VALUE -
6490        ! SO GO ON TO NEXT I VALUE.
6500 !
6510   IF Ictr(X)>0 THEN GOTO Contl
6520 !
6530 Next_x:  NEXT X
6540 !
6550 ! *************************************************************************
6560 !
6570 ! IF THE DETAILED PRINTOUT OPTION HAS BEEN SELECTED THEN PRINTOUT
6580 !       THE F MATRIX FOR THIS CASE
6590 !
6600 ! *************************************************************************
6610 !
6620  IF Prt_opt$<>"S" THEN GOTO 6720
6640  IF Number>=24 THEN PRINT PAGE
6645  PRINT LIN(3)
6650  PRINT "                            UNREDUCED F MATRIX"
6660  PRINT
6670  PRINT "                                   g2a(D,E) VALUE   "
6680  PRINT "                 I VALUE       0    1    2    3   "
6690  FOR X=0 TO 7
6700        PRINT USING "18X,D,8X,K,K,K,K";X,F$(X,0),F$(X,1),F$(X,2),F$(X,3)
6710  NEXT X
6720  SUBEND
6730 !
6740 ! *************************************************************************
6750 !
6760 !                      REDUCE SUBROUTINE
6770 !
6780 ! NOW ALL THE I VALUES HAVE BEEN EVALUATED AND THE F MATRIX OF
6790 ! ALL THE REQUIRED INPUTS HAS BEEN ASSIGNED FOR THIS PERMUTATION.
6800 ! WE MUST NOW COMPARE THESE F MATRIX VALUES TO SEE IF WE CAN
6810 ! REDUCE THE REQUIRED NUMBER OF MULTIPLEXERS NEEDED TO REALIZE
6820 ! THIS CASE.
6830 !
6840 ! *************************************************************************
6850 !
6860  SUB Reduce(INTEGER Where)
6870  OPTION BASE 0
6880  COM Orig_ele(*),Orig_array(*),Bin_equiv(*),Number
6890  COM Min_mult,Min_i_inp$(*),Mult,Min_where
6900  COM Perm(*),Alpha_perm$(*)
6910  COM Work_array(*),F$(*),I(*),Ictr(*),Point(*),Tempf$(*)
6920  COM Zero$,One$,Defalt$,Inp_f$,Inp_nf$
6930  COM A111$,A110$,Equiv$,Comple$,Numb$(*),Prt_opt$,Print_option
6940  COM Temp_array(*),Dividend(*)
6950  COM Min_work(*),Min_temp(*),Min_i(*)
6960  INTEGER X,Z,Blanks,Y
6970 !
6980 ! *************************************************************************
6990 !
7000 !  Z CONTROLS THE COUNTER THROUGH ALL THE I VALUES OF THE F MATRIX
```

```
7010 !
7020 !  WE CAN REDUCE IF: 1) ALL F INPUTS FOR AN I VALUE ARE EQUAL TO 1
7030 !                    2) ALL F INPUTS FOR AN I VALUE ARE EQUAL TO 0
7040 !                    3) F INPUTS FOR TWO DIFFERENT I VALUES ARE IDENTICAL
7050 !                    4) F INPUTS FOR TWO DIFFERENT I VALUES ARE COMPLEMENTS
7060 !
7070 ! **************************************************************************
7080 !
7090  Mult=8
7100  FOR Z=0 TO 7
7110 Ckone:  ! CHECK IF ALL F INPUTS FOR THIS I VALUE ARE = 1.
7120         ! IF YES, THEN WE CAN REDUCE THE NUMBER OF REQUIRED
7130         ! MULTIPLEXERS BY 1.
7140         FOR X=0 TO 3
7150              IF F$(Z,X)<>One$ THEN GOTO Ckzero
7160         NEXT X
7170         ! HERE ALL F INPUTS = 1  -  REDUCE MULTIPLEXERS COUNTER
7180         Mult=Mult-1
7190         F$(Z,0)=All1$
7200         Blanks=3
7210         GOTO Clearf
7220 Ckzero: ! CHECK IF ALL F INPUTS FOR THIS I VALUE ARE = 0.
7230         ! IF YES, THEN WE CAN REDUCE THE NUMBER OF REQUIRED
7240         ! MULTIPLEXERS BY 1.
7250         FOR X=0 TO 3
7260              IF F$(Z,X)<>Zero$ THEN GOTO Ckcont
7270         NEXT X
7280         ! HERE ALL F INPUTS = 0  -  REDUCE MULTIPLEXERS COUNTER
7290         Mult=Mult-1
7300         F$(Z,0)=All0$
7310         Blanks=3
7320         GOTO Clearf
7330 Ckcont: ! CONTINUE COMPARING EACH SET OF INPUTS FOR EACH I VALUE
7340         FOR Y=7 TO Z+1 STEP -1
7350 Ckeqv:          ! CHECK IF THESE TWO ARE EQUIVALENT
7360                 ! IF YES, THEN REDUCE THE NUMBER OF MULTIPLEXERS BY 1.
7370                 FOR X=0 TO 3
7380                      IF F$(Z,X)<>F$(Y,X) THEN GOTO Ckcomp
7390                 NEXT X
7400                 ! HERE THESE TWO I'S INPUTS ARE EQUIVALENT  -  REDUCE.
7410                 ! PUT MESSAGE UNDER Z'S I VALUE AND ELIMINATE THIS
7420                 ! MULTIPLEXER.
7430                 Mult=Mult-1
7440                 F$(Z,0)=Equiv$
7450                 F$(Z,3)=Numb$(Y)
7460                 Blanks=2
7470                 GOTO Clearf
7480 Ckcomp:         ! CHECK IF THESE TWO ARE COMPLEMENTS
7490                 ! IF YES, THEN REDUCE THE NUMBER OF MULTIPLEXERS BY 1.
7500                 FOR X=0 TO 3
7510                   IF (F$(Z,X)=One$) AND (F$(Y,X)<>Zero$) THEN GOTO Next_y
7520                   IF (F$(Z,X)=Zero$) AND (F$(Y,X)<>One$) THEN GOTO Next_y
7530                   IF (F$(Z,X)=Inp_f$) AND (F$(Y,X)<>Inp_nf$) THEN GOTO Next_y
7540                   IF (F$(Z,X)=Inp_nf$) AND (F$(Y,X)<>Inp_f$) THEN GOTO Next_y
7550                 NEXT X
```

```
7560                ! HERE THESE TWO I'S VALUES ARE COMPLEMENTS  -  REDUCE.
7570                ! PUT MESSAGE UNDER Z'S I VALUE AND ELIMINATE THIS
7580                ! MULTIPLEXER.
7590                Mult=Mult-1
7600                F$(Z,0)=Comple$
7610                F$(Z,3)=Numb$(Y)
7620                Blanks=2
7630 Clearf: ! HERE WE HAVE REDUCED - CLEAR REST OF F MATRIX FOR THIS I VALUE
7640         FOR X=1 TO Blanks
7650              F$(Z,X)=" "
7660         NEXT X
7670         GOTO Next_z
7680 Next_y: NEXT Y    ! SO FAR THIS IS A UNIQUE I VALUE
7690                   ! IF WE REACH THE END OF THIS Y LOOP THEN THIS IS A
7700                   ! UNIQUE I VALUE AND THE F MATRIX REMAINS AS ASSIGNED.
7710 Next_z: NEXT Z    ! CHECK NEXT I VALUE
7720 !
7730 ! ************************************************************************
7740 !
7750 ! IF THE DETAILED PRINTOUT OPTION IS SELECTED THEN PRINT OUT
7760 !       THE REDUCED F MATRIX FOR THIS CASE
7770 !
7780 ! ************************************************************************
7790 !
7800  IF Prt_opt$<>"S" THEN GOTO 8070
7810  PRINT LIN(3)
7820  PRINT "                       REDUCED F MATRIX       MULTIPLEXERS=";Mul
t
7830  PRINT
7840  PRINT "                                  g2a(D,E) VALUE    "
7850  PRINT "              I VALUE       0    1    2    3    "
7860  FOR X=0 TO 7
7870       PRINT USING "18X,D,8X,K,K,K,K";X,F$(X,0),F$(X,1),F$(X,2),F$(X,3)
7880  NEXT X
7890 !
7900 ! ************************************************************************
7910 !
7920 ! CHECK IF THIS PERMUTATION'S REDUCED F$ MATRIX REQUIRES LESS
7930 ! MULTIPLEXERS THAN THE MINIMUM TO DATE. IF YES, STORE THE
7940 ! INFORMATION FOR THIS CASE.
7950 !
7960 !   MIN_WHERE = MINIMUM PERMUTATION NUMBER
7970 !    MIN_MULT = MINIMUM NUMBER OF MULTIPLEXERS REQUIRED
7980 !   MIN_I_INP = REDUCED F MATRIX INPUTS FOR MINIMUM
7990 !    MIN_TEMP = WORKING_ARRAY OF MINIMUM BEFORE SORTED
8000 !    MIN_WORK = WORKING_ARRAY OF MINIMUM AFTER SORTED
8010 !       MIN_I = I VALUES OF THE REORDERED MINTERMS FOR THE MIMINUM
8020 !
8030 ! RETURN TO GET THE NEXT PERMUTATION.
8040 !
8050 ! ************************************************************************
8060 !
8070  IF Mult>=Min_mult THEN GOTO 8240
8080  !  THIS IS A NEW MINIMUM
8081  PRINT LIN(2)
```

```
8090  PRINT USING "#,43X,K,K";"THIS IS A"," NEW MINIMUM:"
8100  PRINT USING "3X,6(K)";Alpha_perm$(Where,5),Alpha_perm$(Where,4),Alpha_perm
$(Where,3),Alpha_perm$(Where,2),Alpha_perm$(Where,1),Alpha_perm$(Where,0)
8110  PRINT
8120  PRINT USING "43X,K,3D";"MULTIPLEXERS = ";Mult
8130  Min_where=Where+1
8140  MAT Min_temp=Temp_array
8150  MAT Min_work=Work_array
8160  MAT Min_i=I
8170  FOR X=0 TO 7
8180       FOR Y=0 TO 3
8190            Min_i_inp$(X,Y)=F$(X,Y)
8200       NEXT Y
8210  NEXT X
8220  Min_mult=Mult
8230  WAIT 2000
8240  SUBEND
8250 !
8260  END
```

THIS PAGE INTENTIONALLY LEFT BLANK

END

FILMED

1-86

DTIC